2D and 3D Digital Shape Modelling Strategies

Silvia Rueda López, BEng., MSc.

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy.

June 2010

To Sergio,
my parents,
and my grandparents,
with love.

– Sylvia

Abstract

Image segmentation of organs in medical images using model-based approaches requires a priori information which is often given by manually tagging landmarks on a training set of shapes. This is a tedious, time-consuming, and error prone task. To overcome some of these drawbacks, several automatic methods were devised. Identification of the same homologous set of points in a training set of object shapes is the most crucial step in Active Shape Modelling, which has encountered several challenges. The most crucial among these are: (C1) defining and characterizing landmarks; (C2) obtaining landmarks at the desired level of detail; (C3) ensuring homology; (C4) generalizing to $n > 2$ dimensions; (C5) achieving practical computations. This thesis proposes several novel modelling techniques attempting to meet C1-C5. In this process, this thesis makes the following key contributions: the concept of local scale for shapes; the idea of allowing level of detail for selecting landmarks; the concept of equalization of shape variance for selecting landmarks; the idea of recursively subdividing shapes and letting the sub-shapes guide landmark selection, which is a very general n-dimensional strategy; the idea of virtual landmarks, which may be situated anywhere relative to, not necessarily on, the shape boundary; a new compactness measure that considers both the number of landmarks and the number of modes selected as independent variables.

The first of three methods uses the c-scale shape descriptor, based on the new concept of curvature-scale, to automatically locate mathematical landmarks on the mean of the training shapes. The landmarks are propagated to the training shapes to establish correspondence

among shapes. Since all shapes of the same family do not necessarily present exactly the same shape features, another novel method was devised that takes into account the real shape variability existing in the training set and that is guided by the strategy of equalization of the variance observed in the training set for selecting landmarks. By incorporating the above basic concepts into modelling, a third family of methods with numerous possibilities was developed, taking into account shape features, and the variability among shapes, while being easily generalized to the 3D space. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at a higher resolution. The best strategy to use within the family will have to be determined according to the clinical application at hand.

All methods were evaluated in terms of compactness on two data sets - 40 CT images of the liver and 40 MR images of the talus bone of the foot. Further, numerous artificial shapes with known salient points were also used for testing the accuracy of the proposed methods. The results show that, for the same number of landmarks, the proposed methods are more compact than manual and equally spaced annotations. Besides, the accuracy (in terms of false positives and negatives and the location of landmarks) of the proposed shape descriptor on artificial shapes is considerably superior to a state-of-the-art scale space approach to finding salient points on shapes.

Publications

The publications in journals and peer-reviewed international conferences that resulted from this thesis work to date are listed below.

[1] **S. Rueda**, J. K. Udupa, and L. Bai. Shape Modeling Via Local Curvature Scale, 20^{th} SIBGRAPI: Advances in Image Processing and Computer Vision, Pattern Recognition Letters, Volume 31, pp.324-336, 2010.

[2] **S. Rueda**, J. K. Udupa, and L. Bai. A comparison of local and global scale approaches in characterizing shapes. Proceedings of SPIE Medical Imaging 2009: Image Processing, Volume 7259, pp.725951-1–725951-12, 8^{th}-12^{th} February 2009, Orlando, FL, USA. (Cum Laude Best Poster Award)

[3] **S. Rueda** and J. K. Udupa. nD Statistical Shape Model Building via Recursive Boundary Subdivision. Proceedings of SPIE Medical Imaging 2009: Visualization, Image-guided Procedures, and Modeling, Volume 7261, pp.72611I-1–72611I-11, 8^{th}-12^{th} February 2009, Orlando, FL, USA.

[4] **S. Rueda**, J. K. Udupa, and L. Bai. Curvature and shape variance based landmark tagging methods for building statistical object models. Proceedings of SPIE Medical Imaging 2009: Visualization, Image-guided Procedures, and Modeling, Volume 7261, pp.726129-1–726129-12, 8^{th}-12^{th} February 2009, Orlando, FL, USA.

[5] **S. Rueda**, J. K. Udupa, and L. Bai. Shape Description via local curvature scale.

Proceedings of the Marie Curie Conference, ESOF 2008, P-102, pp.38, 17^{th}-18^{th} July 2008, Barcelona, Spain. (Abstract only)

[6] **S. Rueda**, J. K. Udupa, and L. Bai. Landmark Selection for Shape Model Construction Via Equalization of Variance. IEEE International Symposium on Biomedical Imaging (ISBI) 2008: from Nano to Macro, pp.1271-1274, 14^{th}-17^{th} May 2008, Paris, France.

[7] **S. Rueda**, J. K. Udupa, and L. Bai. A new method of automatic landmark tagging for shape model construction via local curvature scale. Proceedings of SPIE Medical Imaging 2008: Visualization, Image-guided Procedures, and Modeling, Volume 6918, pp.69180N-1–69180N-12, 16^{th}-21^{st} February 2008, San Diego, CA, USA.

[8] **S. Rueda**, J. K. Udupa, and L. Bai. Local curvature scale: a new concept of shape description. Proceedings of SPIE Medical Imaging 2008: Image Processing, Volume 6914, pp.69144Q-1–69144Q-11, 16^{th}-21^{st} February 2008, San Diego, CA, USA.

[9] **S. Rueda**, J. K. Udupa, and L. Bai. Local Scale for Boundary Shape Description: Application in Locating Landmarks Automatically. IEEE CS Press, Proceedings of the 20th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007), pp.313-320, Belo Horizonte, Brazil, October 2007.

Presentations

During this three-year period, I have disseminated the results of my research through the following presentations and seminars:

- **Novel Strategies for describing 2D and 3D shapes**, contributed talk at the Marie Curie Workshop on Medical Image Analysis, University of Nottingham, U.K., 15^{th} September 2009.

- **Novel Strategies for describing 2D shapes**, seminar at the Medical Image Processing Group (MIPG), Department of Radiology, University of Pennsylvania, USA, 9^{th} March 2009 (Invited)

- **nD statistical shape model building via Recursive Boundary Subdivision**, contributed talk at the SPIE Medical Imaging, Conference on Visualization, Image-guided Procedures, and Modeling, Orlando, FL, USA, 10^{th} February 2009.

- **Curvature-Scale and Variance Equalization for Landmark Tagging for Object Shape Modeling**, contributed talk at the Workshop on Computing Intensive Medical Image Analysis, Port d'Informacio Cientifica (PIC) Grid Centre (Universitat Autonoma de Barcelona), Barcelona, Spain, 16^{th} July 2008. (Invited)

- **Landmark selection for shape model construction via equalization of variance**, contributed talk at the IEEE International Symposium on Biomedical Imaging (ISBI): from Nano to Macro, Paris, France, 17^{th} May 2008.

- **A new method of automatic landmark tagging for shape model construction via local curvature scale**, contributed talk at the SPIE Medical Imaging, Conference on Visualization, Image-guided Procedures, and Modeling, San Diego, USA., 17^{th} February 2008.

- **Local scale for boundary shape description: application in locating landmarks automatically**, contributed talk at the 20th Brazilian Symposium on Computer Graphics and Image Processing, Belo Horizonte, Brazil, 10^{th} October 2007.

- **Intrinsic Boundary Shape Features: Theory and Use in Automatically Tagging Landmarks**, contributed talk at the Marie Curie Workshop on Medical Image Analysis, University of Nottingham, U.K., 12^{th} July 2007.

Acknowledgements

This has been by far the most intense professional experience I lived to-date, especially the challenging part of writing the thesis. I would like to thank those who contributed to the achievement of this thesis in many different ways, and particularly, those who touched my life during this three-year period and gave me the support, care, and love I needed to make this happen.

First of all, I would like to thank my supervisors Dr. Tony Pridmore, from the University of Nottingham, and Prof. Jayaram K. Udupa, from the University of Pennsylvania, Philadelphia. Dr. Pridmore provided me with support, understanding, and useful comments along this period, and I am grateful to him for making things easier for me. I cannot start to thank Prof. Udupa for his close supervision, constant support, patience, and friendship throughout these years. Prof. Udupa acted as a mentor to me, by constantly sharing his wisdom, enthusiasm, and ideas with me, while guiding my research career with elegance, scientific rigour, methodology, and professionalism. I am also grateful to him for enriching my doctoral experience and inviting me to his group as a visiting scholar for a 4-month period. I learnt a great deal from the visit and encountered wonderful people in his group, such as Mary Blue, George Grevera, Bruce Hirsch, Samuel Matej, Xinjian Chen, Xiaofen Zheng, Dewey Odhner, and Chris Ciesielski, to name a few.

I would like to acknowledge the European Commission FP6 Marie Curie Action Pro-

gramme (MEST-CT-2005-021170) for funding this research, and to its coordinator Dr. Li Bai. The work has been done in a collaborative framework between the University of Nottingham and the Medical Image Processing Group (MIPG), supported by the EPSRC Visiting Fellowship in Advanced Medical Image Processing and Analysis (EP/F01368X/1) awarded to Professor Jayaram K. Udupa, Chief of the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, USA, for a period of three years. I also wish to thank Google for selecting me as the recipient of the Google Anita Borg Scholarship 2008, which covered some of my research expenses. Many thanks to the University of Nottingham for funding my 4-month secondment at the Medical Image Processing Group, USA. I am grateful to Prof. Steve Benford and Karen Attwood for making this happen and for their help during this period. The medical images used in this thesis were provided by the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, USA.

None of this would have been possible without the unconditional support from friends and family. Special thanks to Ulas, Amalia, Yongnan, Aaron, Stamatis, Diwei, etc. for their friendship and kindness during these years. I would also like to thank my friends from Quarry Quilters, Sue, Karen, Linda, Doreen, Christine, Jill, Avis, Shona, and Gail, with whom I shared unforgettable and heart-warming evenings. Special thanks to Sue and David for their love and for welcoming me into their family. Huge thanks to Julie Robinson, who kept me healthy with her wonderful yoga classes. I would also like to thank my friends Coco and Credo for their love during these years and for bringing to my life their beautiful daughter Ainhoa, who I love deeply. Thanks as well to all friends who gave me company from a distance: Ana, Encarna, Mari Carmen, Ana Maria, Catalina, Diana, Laura, Ainhoa, Carmen, Violette, and Naji, to name a few. Many thanks to Sarah and Sana for their support and love.

Special thanks to my family, especially my parents Juan and Caty, my grandparents Alejandro and Trinidad, my little brother Yuri, my aunt Aida, my uncle Daniel, and my cousins Eva and Bárbara, for being always there for me. Thanks for their understanding of all the

moments I missed and that I could not share with them, for their sacrifice of not having me closer, and for their unconditional support and love. I also wish to thank my grandparents Antonio and María del Carmen for their love and for making my life better.

Very special thanks to Sergio, the one and only, who I love deeply, for taking very good care of me during this period of time, for his unconditional love, support, understanding, humour, care, patience, and for encouraging me to follow my dreams everyday of my life. I could not have done it without him and all the help and love he provided me with during these past years.

I also wish to thank Prof. Alison Noble and Dr. Julia Schnabel for giving me a good reason to finish this thesis.

And last but not least, I would like to thank the examiners of this thesis, Dr. Guoping Qiu and Prof. Alexandre Falcão, for their useful comments and suggestions.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

When you are describing
*A **shape**, or sound, or tint;*
Don't state the matter plainly,
But put it in a hint;
And learn to look at all things
With a sort of mental squint.

– Lewis Carroll

1.1 Motivation and Rationale

Clinical radiology has witnessed a substantial evolution during the past decades, with the introduction of digital images into the medical environment. As a consequence, there has been a considerable increase in the volume of data that needs to be processed accompanied by an extensive image quality improvement. The need for automated computerized systems is now a reality for the analysis and visualization of medical images. One of the critical operations needed in most image processing systems is segmentation. Segmentation is a two step process, commonly and extensively used in the medical field, that consists of identifying the whereabouts (recognition) and defining the spatial occupation (delineation) of objects in images.

Segmentation of organs in medical images is a difficult task intrinsically related to the shape of the object of interest. Several approaches are available at present for modelling *a priori* object shape information. Among these, landmark based statistical modelling tech-

niques, such as Active Shape Modelling (ASM) [Cootes et al., 1995], are the most extant and have proven useful in a variety of medical image processing and analysis operations. Shape models facilitate automatic object recognition, delineation, registration, and shape analysis. To build the model, we need a training set of shape examples annotated with corresponding landmarks. Manual positioning of landmarks on training shapes/images is a tedious, subjective, time-consuming, and error prone task, and almost impossible in the $3D$ space. To overcome the drawbacks of manually creating a training set, the development of automatic methods has been actively pursued. Automatic annotation should capture the real variability existing among shapes with landmark correspondences expressed among shapes of the training set. Identification of the same homologous set of points in a training set of object shapes is the most crucial step in ASM, which has encountered challenges, the most crucial among these being (C1) mathematically defining and characterizing anatomical landmarks; (C2) obtaining landmarks at the desired level of detail; (C3) ensuring homology among landmarks; (C4) generalizing methods to $n > 2$ dimensions; (C5) achieving practical computations.

This thesis focuses on the creation of $2D$ and $3D$ models and the study of shape. There are major gaps and drawbacks in the current shape modelling techniques. This thesis proposes several novel modelling techniques that try to fill in some of these gaps by developing robust, automatic landmark tagging methods for model-based segmentation approaches while attempting to meet C1-C5.

The initial focus is on $2D$ shapes, using their shape characteristics as defined by prominent points with high and/or low curvature as potential landmarks. However, the methods of estimation of curvature, existing in the literature, were found not accurate and robust enough when applied to digital boundaries. Therefore, new methods needed to be investigated to express the curvature of digital contours by using the concept of local scale. By using this curvature calculation, a comprehensive shape descriptor was developed, which was able to extract dominant points at different levels of detail naturally, possessing many other useful shape properties.

By using the shape descriptor thus created, an automatic landmark tagging approach was devised based on the extraction of dominant points on the mean shape of a training set of shapes. These points were propagated to all the shapes of the training set to establish correspondence among shapes. Since all the shapes of a same family do not necessarily present exactly the same shape features among shapes, which may often weaken methods based on dominant points, we argued that, to adequately represent shape variability, landmark selection should be tied to the variability existing among training shapes. Therefore, we devised a novel method that takes into account the real variability existing in the training set of shapes and that places more landmarks where there is more variability, to create a shape model that really characterizes the variability of that family of shapes.

By incorporating the above basic concepts into modelling, we devised a family of automatic landmark tagging methods with numerous possibilities, that takes into account shape features, and the variability among shapes, while being easily generalized to the $3D$ space. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at a higher resolution. The results obtained by the different methods presented will vary depending on the application. Therefore, the best automatic landmark tagging method to use will have to be selected depending on the particular clinical application under consideration.

1.2 Specific Aims

The specific aims of this thesis are as follows:

- To develop an accurate and robust local curvature measure to use for the detection of dominant points on digital contours, giving a comprehensive description of shape at different levels of detail (multi-scale).

- To devise algorithms to automatically identify shape-salient points in digital boundaries. Salient points can be identified by using different techniques such as via high

and/or low curvature points, Principal Component Analysis (PCA), geometrical shape properties, etc.

- To devise techniques to automatically distribute landmarks on each contour to best capture shape variation existing in the whole training set.
- To study how effective these points selected via different strategies may be for use as landmarks in shape models.
- To develop algorithms for automatically constructing shape models by using these points.
- To develop efficient algorithms that can be generalized to the $3D$ space.
- To develop methods that are scalable by level of detail.
- To develop algorithms for assessing the model compactness.

1.3 Structure of the Thesis

The remainder of this thesis is divided into four main parts. The first part describes a novel shape descriptor based on curvature-scale and how it performs on medical and non-medical shapes. The second part introduces novel methods for $2D$ automatic landmark tagging by applying different techniques. The third part attempts to extend one of these novel methods into the $3D$ space. The fourth and last part contains the conclusions and discussion of the main work developed in this thesis. The summary of the content of each chapter is presented below.

Chapter 2 gives the general background of the thesis by reviewing some of the relevant methodologies previously developed in the fields of shape description and automatic landmark tagging.

Chapter 3 describes the mathematical theory of a novel $2D$ digital multi-scale shape descriptor based on local curvature. First, we introduce a method for local curvature extraction, based on a new local scale concept called c-scale. This method returns several factors related to curvature that are important for the interpretation of shapes. We also explain how to use the c-scale value to create a shape descriptor based on local curvature, which will give us a complete description of shape.

Chapter 4 presents a quantitative and qualitative evaluation of the shape descriptor proposed in Chapter 3. The method is applied to medical and non-medical objects, and compared to several shape description strategies that are considered state-of-the-art in this field.

Chapter 5 delineates a framework containing several strategies for $2D$ automatic landmark tagging based on the c-scale shape descriptor introduced in Chapter 3. Several techniques for landmark selection and landmark propagation are presented.

Chapter 6 introduces a novel $2D$ automatic landmark tagging method based on the idea of variance equalization. This method creates a model by taking into account the variability existing in the training set of shapes, by placing more landmarks where the variance among shapes is high, and less where it is low. The distribution of landmarks is achieved by variance equalization.

Chapter 7 proposes a family of novel global-to-local strategies based on the concept of recursive boundary subdivision of shapes simultaneously on all training shapes, maintaining synchrony of the level of recursion, while keeping correspondence among generated landmarks automatically by the correspondence of the homologous shape segments in the training shapes. Several methods are devised to place landmarks on corresponding segments by using different strategies.

Chapter 8 presents a qualitative and quantitative evaluation of the automatic landmark

tagging methods described in Chapters 5-7. The quantitative assessment is performed by newly devised compactness factors. The results are displayed on two medical objects - the talus bone of the foot (extracted from MR images) and the liver (extracted from CT images).

Chapter 9 extends the concepts in Chapter 7 to the 3D space. A preliminary demonstration of the results is presented for a $3D$ object - the talus bone of the foot.

Chapter 10 discusses the methods and results of this thesis and points to some open problems identified during the course of this thesis work.

2

Background

This chapter presents a literature review of two central topics of the thesis: shape description approaches and automatic landmark tagging methods, both necessary to create shape models for model-based image analysis approaches.

2.1 Shape Description

Shape description plays a fundamental role in computer vision and pattern recognition, especially in the fields of automatically defining landmarks, shape analysis, image segmentation, and registration. Many medical applications require a good description of the boundary shape of the structure of interest. A shape descriptor aims at characterizing a shape uniquely while being invariant to rotation, scale, and translation [Dryden and Mardia, 1998]. Shape representations must be unique, complete, and should be able to reflect the differences among similar objects while abstracting from detail and keeping the basic features. From the chosen representation, we should be able to characterize the object and express numerically the associated shape descriptor vector. During the past decades, several shape description methods have been described. Many of them are complex, application dependent, and cannot deal well with digital boundary effects and noise.

In dealing with the computational aspects of shape, it is important to distinguish between two distinct concepts - *shape representation* and *shape description*. We will briefly review the most common techniques for shape representation and focus more on shape descriptors,

our goal being the extraction of dominant points from a contour.

2.1.1 Shape Representation

Prior to shape description, we need to have a proper representation of shape that preserves the important features of the boundary contour. Numerous methods for shape representation exist. According to [Costa and Cesar Jr., 2001], the existing shape representation methods can be classified into contour-based, region-based, and transform-based approaches (Fig. 2.1). All contour-based methods need to extract the boundary contour of the shape from the given shape (which is often given as an image) and represent it in a manner suitable for shape analysis.

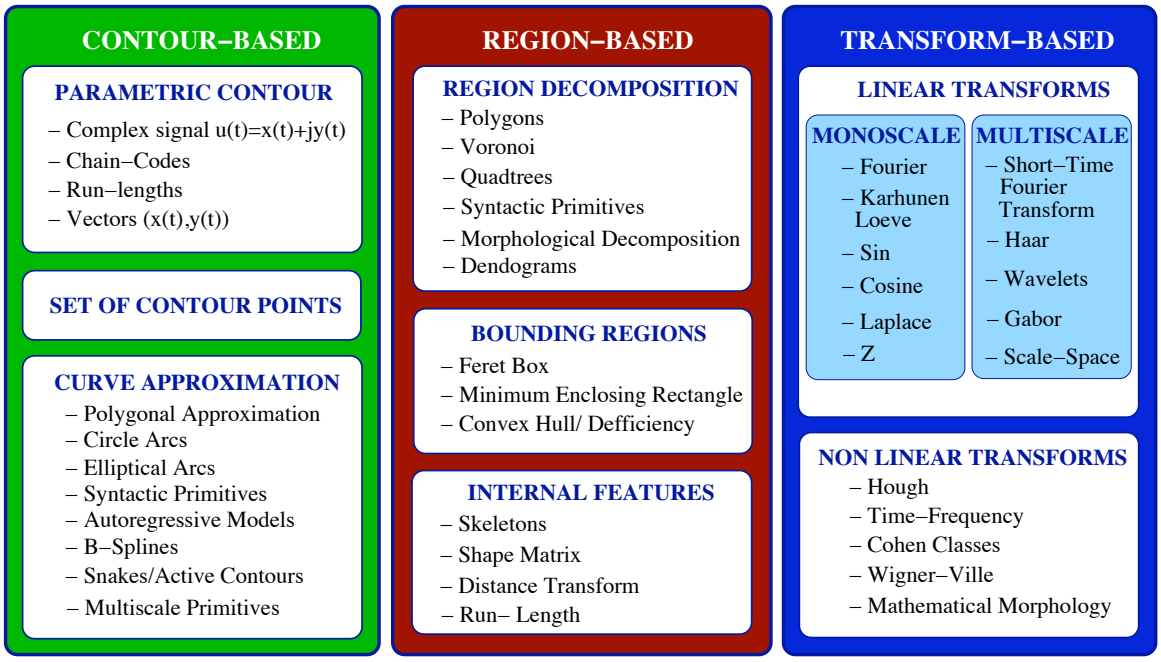


Figure 2.1: A classification of shape representation methods.

The contour-based approaches are divided into three classes: parametric contours, set of contour points without any given order, and curve approximation. A parametric representation of a shape can be obtained after applying some contour following algorithms, such as the ones presented in [Costa and Cesar Jr., 2001]. Once the parametric contour has been extracted, it can be represented by different data structures (e.g. vectors and complex

signals) or by chain codes [Freeman, 1961] [Freeman and Saghri, 1980] [Freeman, 1978]. Another option to represent a boundary is to approximate or interpolate it piecewise by certain primitives. In order to achieve this, the contour has to be segmented and each segment approximated. The primitive that is usually used is a straight line, giving as a result a polygonal approximation of the contour (e.g. Ramer Polygonal Approximation [Ramer, 1972]; Split-and-merge algorithm [Costa and Cesar Jr., 2001]).

In region-based approaches, the shape is represented by the region inside the object. We can distinguish among three different classes: methods based on region decomposition, bounding regions, and internal features. In the first class, a set of primitives is used to represent the shape region. In the bounding regions approach, we approximate the shape by a pre-defined geometric primitive. We can also use certain features of the internal region of the shape to represent it.

The main transform-based approaches, divided into linear and non linear transforms, are used for both shape representation and description. In this family of methods, a shape is represented by its transformation in a monoscale or multiscale manner. From these representations, we can extract domain descriptors such as transform coefficients, transform measures, and transform statistics.

In our case, we do not want to approximate the contour by any method. We would like to work with the exact digital contour obtained after extracting the boundary of a segmented object. Consequently, in this study, we have extracted the boundary of the object, and represented it by using the vector coordinates $(x(t), y(t))$ for each pixel t forming the contour. We consider the contour to be of one-pixel width. Therefore, we are using a contour-based approach, obtaining a parametric contour described by vectors. We will use this real contour to characterize the shape of the object through shape descriptors that attempt to define the dominant points of the shape. Any other method will approximate the digital contour, which may suppress some of the information that may be useful for the extraction of these

dominant points.

2.1.2 Shape Description/Characterization

Shape characterization refers to the processes that yield in a numeric description of the shape. The goal of shape characterization is to obtain a simplified representation of the original shape while retaining its most important features. What we consider an important feature will differ depending on the application. In general, a shape description method generates a feature vector from a given shape and aims at characterizing a shape uniquely while being invariant to rotation, scale, and translation [Dryden and Mardia, 1998]. The shape must not be altered by these transformations. It is worth stressing that, normally, a shape descriptor works as a pair in the sense that features need to be extracted, and then, similarities between shapes need to be computed. These similarity metrics do not need to be invariant to rotation, translation, and scale. However, the scope of this thesis is to tackle the feature extraction part only and will not compare shapes with similarity metrics, as our purpose is not to do image retrieval but to extract dominant points to characterize shapes.

Shape description methods can be mainly classified into *contour-based* and *region-based* approaches [Zhang and Lu, 2004] [Loncaric, 1998]. The contour-based class uses the shape boundary whereas region-based methods consider the inside of the shape to extract the features of interest. Although many methods for shape description exist, there is none that could be labelled as a general method, in the sense that they are usually application dependent.

During the past decades, several shape description methods have been put forth for the extraction of dominant points. Most common techniques consist of shape signatures [Davies, 1997] , boundary moments [Sonka et al., 1999], polygonal and curve decomposition [Ramer, 1972][Grosky and Mehrotra, 1990][Berreti et al., 2000], syntactic analysis [Fu, 1982], scale space analysis [Witkin, 1983] [Asada and Brady, 1986] [Ray, B.K and Ray, K.S, 1995] [Mokhtarian et al., 2003] [Rattarangsi and Chin, 1992] [Lindeberg, 1994], spectral transform

(eg. Fourier descriptors [Zahn and Roskies, 1972][Chellapa and Bagdazian, 1984] and wavelet descriptors [Tieng and Boles, 1997]), and defining shape invariants using boundary primitives [Squire and Caelli, 2000]. Most recently, more innovative approaches using graph-based theory for multiscale shape analysis such as the one described by [Torres et al., 2004], which is based on the image foresting transform [Falcão et al., 2004], and a tensor scale method to detect shape saliences [Miranda et al., 2005] [Andaló et al., 2007] have also been developed.

Most of the shape descriptors are complex, application dependent, and cannot deal well with digital boundary effects and noise. A measure of curvature forms the basis of most methods of shape characterization. Some methods deal with it directly, others indirectly. Curvature has been widely used in the past for location of landmarks, as it appears to be the natural and intuitive approach to characterize a family of shapes [Attneave, 1954]. Curvature is one of the most important features used in characterizing shapes, and it allows the extraction of dominant points such as corners and inflection points. It plays an important role in identifying different primitives. The curvature of a smooth curve is defined as the inverse of the radius of its osculating circle at each point (Fig. 2.2).

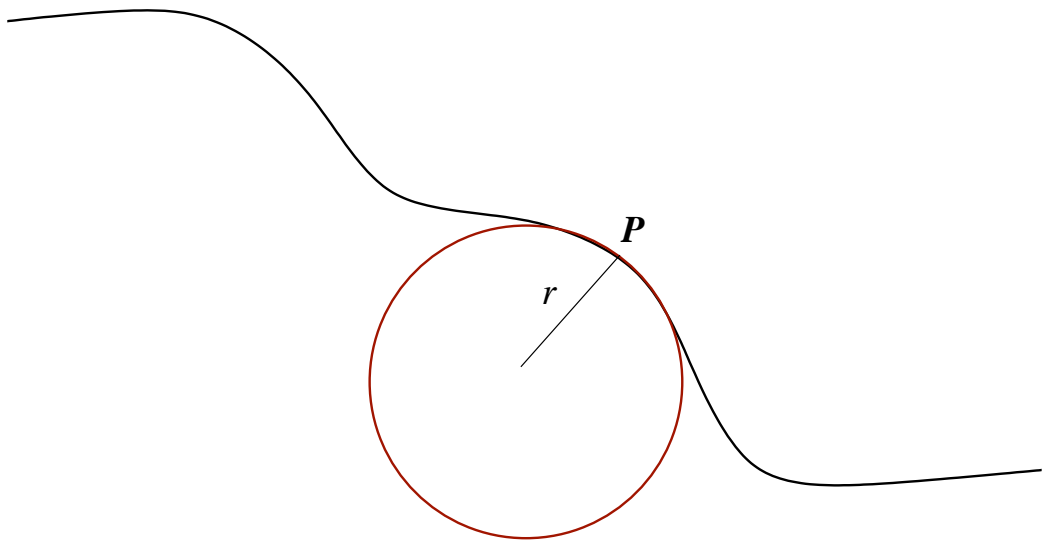


Figure 2.2: The curvature at P is $1/r$, where r is the radius of the osculating circle at P.

Although curvature estimation from a digital contour would seem rather simple, not many methods exist that are easy to implement, fast, and robust to noise and digitization effects. A local absolute maximum in the curvature corresponds to a generic corner in the shape. If this maximum is positive, then, the corner is convex, and if it is negative, the corner is concave. Straight line segments are represented by constant zero curvature. Circular segments correspond to constant non-zero curvature. Zero crossings of the curvature function represent inflection points in the shape. Therefore, just with the feature of curvature, we can potentially fully describe the shape and extract its dominant points as well as its associated characteristics.

Let the shape of a 2D object be represented by its boundary $\mathcal{B}$ in the usual digital space. The elements b_i of $\mathcal{B}$ will be called *bels* (an abbreviation for "boundary elements") or *points* from now on. A bel may be a pixel on the boundary or a pixel edge. In either case, the central points (x, y) coordinates constitute the bel's representation. To estimate the curvature of a digital boundary, five main approaches have been used [Worring and Smeulders, 1992]: orientation-based, derivative-based, osculating circle-based, angle-based, and curvature scale space (CSS) based.

In orientation-based methods, the curvature is estimated by the change of slope of the tangent at each point of the boundary in a certain direction by using chain codes to approximate the boundary and different tangent estimations. The derivative-based approach involves calculating the curvature based on derivatives. To achieve this, the boundary is approximated by second order curves such as splines to obtain a parametric definition of the curve. Then, the curvature is estimated by using

$$\kappa = \frac{x''y' - x'y''}{(x'^2 + y'^2)^{\frac{3}{2}}}. \tag{2.1}$$

In this case, the curvature is calculated on sampled points of the approximated boundary, and the results should be mapped again to the digital boundary using interpolation or aver-

aging [Hermann, 2007] . For the osculating circle approach, after smoothing, the boundary is fitted by a circular disc of a certain radius. Then, the curvature is computed using the inverse of these radii at each bel. To avoid digital effects, in all these curvature estimators, some continuous approximation is attempted when estimating the tangents at the boundary, or when the boundary is approximated to obtain a parametric definition, or when a circle is fitted to a smoothed boundary. This may introduce errors in the boundary or even in the shape.

The angle-based approach [Rosenfeld and Johnston, 1973] is widely used in the literature [Ray, B.K. and Ray, K., 1992] [Neumann and Teisseron, 2002] [Carmona et al., 2005] [Davis, 1977] for extracting significant curvature maxima and minima directly on digital curves. The curvature at each bel b_i is estimated by considering the k-cosine $\mathbf{c}_{ik}$ of the angle formed by the vectors $\mathbf{A}_{ik} = \overrightarrow{b_i b_{i-k}}$ and $\mathbf{B}_{ik} = \overrightarrow{b_i b_{i+k}}$, where $-1 \leq \mathbf{c}_{ik} \leq 1$. $\mathbf{c}_{ik}$ is close to 1 when the curve is turning rapidly, and close to -1 when the curve is relatively straight. The curvature estimation depends on the value of k. Therefore, an appropriate value of k at each point in the curve needs to be found. Certain rules are defined to select the optimal values to detect the local curvature maxima and minima. The main drawback of this method is that it cannot extract all curvature maxima/minima. Also, it does not allow a "level-of-detail"control.

Global approaches, such as CSS [Rattarangsi and Chin, 1992] [Mokhtarian et al., 2003] [Ray, B.K and Ray, K.S, 1995][Ray and Pandyan, 2003][Lindeberg, 1994], convolve the parametric boundary functions in both x and y with a varying scale Gaussian smoothing function. For each scale, the maxima of absolute curvature of the boundary are obtained and the scale space representation is established. The scale space map is transformed into a set of trees [Rattarangsi and Chin, 1992] to simplify the representation and to detect relevant dominant points. To build the trees, it is necessary to know the *persistence*, *sign*, and *movement* for each maximum of curvature detected on the boundary [Rattarangsi and Chin, 1992]. Persistence and movement can be established by tracking the patterns on the scale space map. Once the trees are built, corners are identified at multiple scales in a coarse-to-fine manner.

The selection of corners is based on a set of rules that attempts to assure a stability criterion. The stability is determined by measuring the length of corresponding roots, branches, and leaves of the trees, excluding the absolute maxima caused by digital effects and noise. The main idea of these type of methods is to detect dominant points at different scales and to use the information in all the scales as a whole, to decide whether a point is a corner or not. This method is the state-of-the-art shape descriptor used in the MPEG-7 standard. However, its implementation is subjected to many rules and the results are highly dependent on the selection of the scale at which dominant points are detected.

Global and Local Scale

It is desirable to be able to detect dominant points at different scales so that it becomes possible to vary the level of detail depending on the application. In image processing, the concept of scale evolved from the need to handle variable object size in different parts of the scene. Global scale methods process the scene at each of various fixed scales and combine the results, as in scale space approaches [Mokhtarian et al., 2003] [Lindeberg, 1994] [Rattarangsi and Chin, 1992][Ray, B.K and Ray, K.S, 1995]. Local and locally adaptive scale approaches [Saha et al., 2000][Madabhushi et al., 2006], such as ball-, tensor-, and generalized scale [Madabhushi et al., 2006], define the largest homogeneous region at each point, and treat these as fundamental units in the scene. A dichotomy similar to the above, encountered in analysing images, must exist for describing shapes also. In this thesis, we generalize the concept of local ball scale used in images to shape boundaries and call it local curvature scale. A comparison between global and local scale shape descriptors will be presented in Chapter 4.

2.2 Automatic Landmark Tagging

Segmentation of organs in medical images is a difficult task requiring very often the use of model-based approaches. These methods are based on the construction of a statistical model that will contain all the features and variability of the object of interest among all its pos-

sible manifestations. Subsequently, the model is used to perform a search on a new image not belonging to the training set. The difference between methods lies in how the model is represented and built, if and how scene intensity texture information is incorporated statistically into the model, how an initial position of the model is specified in the scene, and how the particular shape is found in the scene. Prominent in this group are Active Shape Models (ASMs) [Cootes et al., 1995] [Cootes et al., 1994], Active Appearance Models (AAM) [Cootes et al., 2001], m-Reps [Pizer et al., 2003a] [Pizer et al., 2003b], and atlas-based methods [Christensen et al., 1997] [Gee et al., 1993]. ASM and AAM use landmarks to represent the shape and principal component analysis to capture the major modes of variation in shape observed in the training data sets. ASM considers only the shape variation whereas AAM takes into account also the variation in texture inside the object. m-Reps uses the shape information obtained after applying a medial axis representation on the shape. Atlas-based methods rely on a previously created atlas that captures in it information about the object label, geometry, shape, scene intensity properties, and object assembly architecture defining the relationship among objects. In this thesis, we will focus on landmark-based strategies (a la ASM/AAM) to build shape models.

Shape models facilitate automatic object recognition, delineation, registration, and shape analysis. To build the model, we need a training set of shape examples annotated with corresponding landmarks indicated among shapes. Manual positioning of landmarks is a tedious, subjective, time-consuming, and error prone task, and almost impossible in the 3D space. To overcome the drawbacks of manually creating a training set, it is necessary to perform automatic landmark tagging. Automatic annotation should capture the real variability existing among shapes with landmark correspondences expressed among shapes of the training set.

Automatic landmark tagging for model building consists of two main tasks: accurate *landmark positioning* in one reference shape or in all shapes of the training set (depending on the method used), and establishing *landmark correspondences* among shapes of the training set. These two steps are independent, with many possible approaches for each of them. Some

methods handle both aspects in a tightly coupled manner, however, most of the methods focus on one of the two tasks. Many papers just treat the correspondence part, taking one reference image on which landmarks are positioned manually or by some other means. The order in which these two tasks are accomplished leads to a classification of the existing methods into two groups: *local-to-global* and *global-to-local.* In *local-to-global* approaches, local operations first define landmarks, among which correspondence is subsequently established by global operations. In *global-to-local* methods, global operations, first carried out, lead to localization of landmarks which simultaneously establish correspondence. Most of the methods found in the literature are *local-to-global* in nature.

2.2.1 Local to Global

We first review some of the local-to-global methods existing in the literature. We present the methods distinguishing between those focusing on the landmark positioning task and those centred on the landmark correspondence task.

Landmark positioning

The selection of landmarks is usually achieved manually or automatically using methods to find characteristic features on shapes, such as anatomical or mathematical landmarks [Dryden and Mardia, 1998]. The goal is to represent each shape of the training set with a set of landmarks or key points, to generate a compact Point Distribution Model (PDM) [Cootes et al., 1994] that will best capture the shape variation among the training shapes considered. Therefore, each landmark must be located in the same manner in all the examples of the training set. However, manual selection of landmarks in certain anatomical or biological structures is not always trivial, and the same is true in finding correspondences among landmarks in a non-homogeneous class of shapes as in the case of disease states. Previous methods focused on the task of selecting landmarks on shapes or images, usually to create PDMs for model-based approaches. Baumberg and Hogg [Baumberg and Hogg, 1994] proposed a method to select landmarks automatically on contours of human figures in walking pedestrian images. For each contour of the training set, a reference point is defined at

the lowest position of the point of intersection of the principal axis with the contour. This point is then used as the starting point of a cubic length-wise uniformly spaced B-spline, where the control points are the selected landmarks. Walker et al. [Walker et al., 2002] aimed at automatically training appearance models of human face sequences. They use first and second order normalized Gaussian partial derivatives to locate feature vectors on the first image of a sequence and construct saliency images for different scales at coarse and fine levels. The correspondence is tracked across frames distributing the features evenly across each object under certain constraints of scale and distance. This work demonstrates that coarse scale features are more reliable than fine scale features to obtain landmarks. One of the problems of this method comes to fore when features move or the shape of the object in different frames is not the same; then the correspondence fails. Also, features that are salient in one frame are not necessarily salient in another frame. Another approach was introduced by Rohr [Rohr, 1997, Rohr, 1999] for landmark-based registration. He described differential operators for detecting landmarks using first order partial derivatives on images to avoid instabilities of higher order partial derivatives. Souza and Udupa [Souza and Udupa, 2005] use polygonal approximation of contours to find landmarks in the mean shape of the training set. The landmarks are propagated to all the examples of the training set using closest point propagation to establish correspondence. This last method obtains better results than manual and equally spaced annotations, but the location of the landmarks do not precisely correspond to the dominant points on the shape, since the location of the points is dependant on the location of the initial points.

Landmark correspondence

Other methods of automatic landmark tagging focus more on the correspondence task, and in most of these cases, the selection of landmarks is achieved manually. The correspondence can be established using closest point propagation [Souza and Udupa, 2005], optimization approaches [Davies et al., 2003] such as Minimum Description Length (MDL) strategies [Thodberg, 2003a] [Davies et al., 2002] [Thodberg, 2003b], parameterization, or registration [Frangi et al., 2001]. Bookstein [Bookstein, 1997] studied extensively the use of landmarks

for the statistical study of biological shapes. In his work, initial anatomical landmarks are located manually on the shapes. Thin-plate splines and Procrustes analysis are used to establish correspondence between curves among shapes of the training set. Landmarks are allowed to move along the contours in order to minimize the corresponding bending energy of the splines. Other approaches, such as the one studied by Frangi et al. [Frangi et al., 2001], use registration as a way of establishing correspondence among shapes. In this work, a 3D atlas is built automatically using non-rigid registration in a training set of segmented images. A mean binary volume is computed from the training set of binary images to create a 3D binary atlas representative of the class of shapes considered. The landmarks are found on the shape represented by this binary volume by using the marching cubes algorithm to obtain dense triangulation of the boundary surface. Decimation is applied to retain only the minimum number of nodes (landmarks) necessary to represent the surface. Then, by using 3D elastic registration, based on maximizing the mutual information, the landmarks are propagated to all the shapes of the training set. Hill et al. [Hill et al., 1997] and Brett and Taylor [Brett and Taylor, 2000] describe a pairwise non-rigid correspondence method for 2D and 3D respectively, using a binary tree of merged shapes. They achieve correspondence by matching sparse polygonal representations without using curvature estimations. Landmarks are placed on the mean shape at the root of the tree and propagated to the original training set, corresponding to the leaves of the tree. The charge for registration is the use of the reference shape to establish correspondences. Another popular method is the MDL approach described by [Davies et al., 2002] and [Thodberg, 2003a] among others. This method treats landmark correspondence as an optimization problem. In this type of work, the initial landmarks have to be defined on a shape (in an equally spaced manner or with a priori knowledge of where they should be), and then, the correspondence is optimized by minimizing a certain objective function. This method performs better than the manual method and finds correspondences by reparameterizing each shape of the training set. However, the method is complex, computationally slow, and the results obtained depend on the different steps used to implement it. Furthermore, the algorithm does not make explicit use of known properties of the shapes and sometimes places landmarks in locations that do not seem appropriate for the human

notion of landmark correspondence. It has been shown by Thodberg [Thodberg, 2003b] that incorporating curvature information into an MDL description gives better results than using MDL alone. However, this paper also observed that this will be true only for shapes with low noise, due to the curvature calculations that were used.

2.2.2 Global to Local

In contrast, the global-to-local methods [Rueckert et al., 2001] [Marsland and Twining, 2003] [Vaillant et al., 2004] [Cootes et al., 2008] [Twining et al., 2002] side step the problem of detecting landmarks and use all points on the shape boundary to establish correspondence among points on the training shapes via dense point registration. In this case, models can be built by using unlabelled images. Non-rigid registration is used to deform a reference image into each sample of the training set in order to construct the warp fields. The main drawback of this approach is that shapes cannot be represented by a few landmarks and that it relies completely on registration.

2.3 Conclusion

In this chapter, we have presented the background of the two main topics studied in this thesis, which are shape description strategies and automatic landmark tagging methods, both useful to construct shape models for model-based segmentation approaches. We aim at characterizing shapes uniquely with a good descriptor of the object of interest that can extract dominant points with different level of detail to be used in shape description. Our focus is on contour-based strategies based on curvature that can deal with digital boundaries while being multi-scale. Automatic landmark tagging methods consist of two main steps: landmark positioning and landmark correspondence. Both steps need to be accomplished to create a shape model able to characterize a family of shapes. Automatic landmark tagging methods can be classified into global-to-local, and local-to-global, most of them belonging to the latter class. In the following, we present novel methods for shape description and automatic landmark tagging towards the goal of shape modelling.

3

2D Shape Description Based on Local Curvature Scale

This chapter describes a novel contour-based shape descriptor (cf. Section 2.1.2) based on curvature. The proposed method estimates curvature without modifying the boundary, taking into account the digital effects and noise, and considering local scale in curvature calculation, to obtain a complete description of shape with different levels of detail. Further, a byproduct of the method is that a shape can be represented by dominant points, such as high curvature points, or inflections points. Besides, the method can associate with each dominant point a description of its characteristics (convex, concave, flat), which can be very useful in a number of applications, such as automatic landmark tagging or shape retrieval.

In this chapter, we first present the theoretical aspects of the novel 2D shape descriptor in Section 3.1. In Section 3.2, we present the algorithms. In Section 3.3, we describe a state-of-the-art method of shape description that extracts dominant points at multiple scales. This method will be used as a reference for comparatively evaluating the new approach in Chapter 4.

3.1 Theory of Local Curvature Scale (c-scale)

Let the shape of a 2D object be represented by its boundary $\mathcal{B}$ in the usual digital space. For any point b on a boundary $\mathcal{B}$, we define local curvature scale segment or *c-scale segment*

as follows.

Definition 3.1.1 (c-scale segment)**.** *The* ***c-scale segment*** *at any point b on a boundary $\mathcal{B}$ is the largest set of points of $\mathcal{B}$ connected to b such that no point in the set is farther than t from a line connecting the two end points of the set.*

The "homogeneity of boundary region" is expressed in terms of the boundary "straightness" at every point and subsequently by how the c-scale segments change along the boundary. In other words, the c-scale segment at any point b consists of the set of points with low curvature for that region, the curvature being measured by how much the points deviate from the line connecting the two end points of the set.

Let $b_1, ..., b_q$ be the points (or boundary elements - bels) defining a boundary $\mathcal{B}$, and t the scale considered. For a particular scale t, we will associate with each point $b = b_i$ its

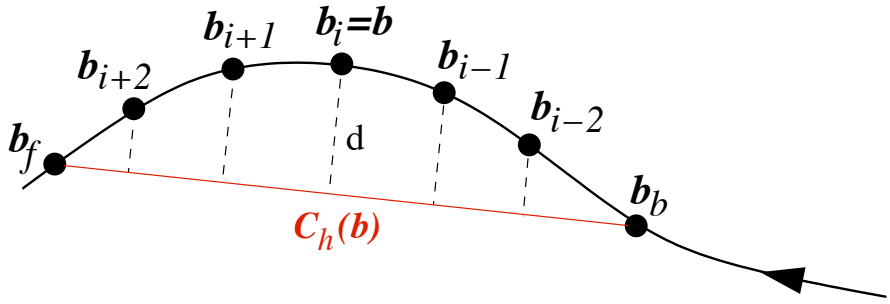


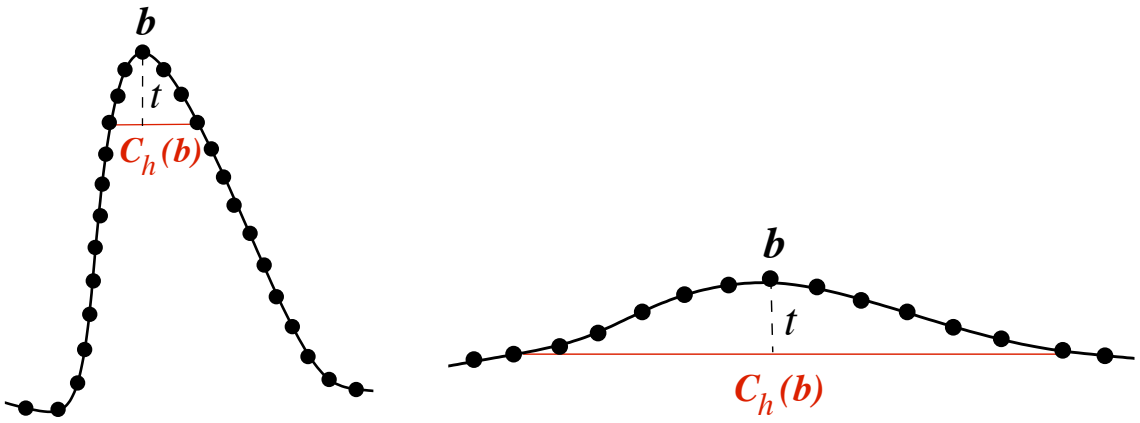
Figure 3.1: c-scale estimation.

c-scale segment $C(b)$. This set, as explained below, is an indirect indicator of the curvature at b. To determine $C(b_i)$, we progressively and symmetrically examine the neighbours, first the set of points b_{i-2}, b_i, b_{i+2}, then the set $b_{i-3}, b_{i-2}, b_i, b_{i+2}, b_{i+3}$, and so on (Fig. 3.1). At each examination, we calculate the distance of the points in the set from the straight line connecting the two end points of the set. If the maximum distance of these points from the line is greater than a threshold t, we define the c-scale segment $C(b)$ of b as the last set of connected points found for which the distance was still within the threshold. There is a *c-scale value* associated with each c-scale segment and defined as follows, which makes it

possible to derive the notion of boundary homogeneity.

Definition 3.1.2 (c-scale value)**.** *The* ***c-scale value*** *assigned to b_i, denoted $C_h(b_i)$, is the chord length corresponding to $C(b_i)$, which is the length of the straight-line segment between the end points b_b and b_f of $C(b_i)$.*

For a certain scale t, if $C_h(b)$ is large, it indicates small curvature at b (Fig. 3.2(b)), and if it is small, it signals high curvature at b (Fig. 3.2(a)).



(a) High curvature at b. (b) Small curvature at b.

Figure 3.2: $C_h(b)$ is an indirect measure of curvature. For the same scale t: (a) $C_h(b)$ is small when there is high curvature at b; (b) $C_h(b)$ is large when there is small curvature at b.

c-scale values are helpful in estimating actual boundary segments and their curvature, independent of digital effects. In the following, we will arrive at the direct relationship existing between the c-scale value $C_h(b)$ and the curvature $\kappa(b)$. We will also derive the relation between $C_h(b)$ and the radius $r(b)$ of the osculating circle at b to the boundary $\mathcal{B}$, and the relation between $C_h(b)$ and the arc length $A(b)$. These relationships associate the curvature, arc length, and radius information of the approximating circle with every element b in the boundary $\mathcal{B}$. c-scale values are represented in terms of arc length.

3.1.1 Relation Between c-scale Value $C_h(b)$ and Radius $r(b)$

From a knowledge of $C_h(b)$ and by assuming that $C(b)$ locally represents a circular arc, we will now arrive at the actual *arc length* $A(b)$ at b corresponding to the c-scale segment $C(b)$.

Let b be a point on a circular arc $\mathcal{A}$. Any segment joining the two ends of the circular arc is a chord P_2P_3 of a circle $\mathcal{C}$ with radius r and centre o (Fig.3.3). The radius and the centre of the circle can be obtained by using the principle of the perpendicular bisector of a chord, which states that *the perpendicular bisector of a chord in a circle $\mathcal{C}$ passes through the centre of the circle.*

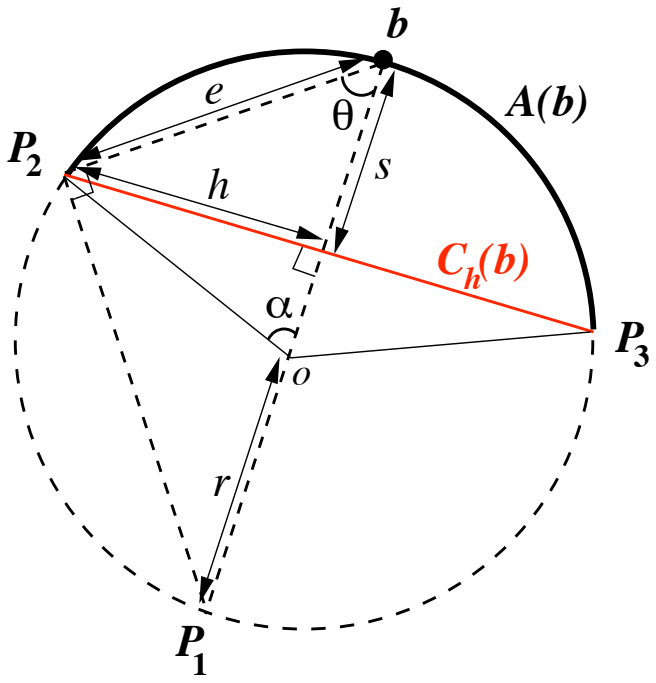


Figure 3.3: Geometric properties of the chord in a circle.

If we trace the perpendicular bisector of the chord, we divide the chord into two equal segments. Let h be the length of one of these segments and s the distance between b and the middle point of the chord. A right triangle of sides h, s, and e can then be defined (Fig.3.3), where

$$e = \sqrt{s^2 + h^2}. \tag{3.1}$$

Further, the triangle P_1P_2b is inscribed in the circle $\mathcal{C}$, and its hypotenuse P_1b is a diameter of the circle. According to Thales theorem, *if P_1, P_2, and b are points on a circle $\mathcal{C}$, with diameter P_1b, then the angle $\widehat{P_1P_2b}$ is a right angle.* Therefore, P_1P_2b is a right triangle.

If we trace the right triangle P_1P_2b, and the right triangle formed by sides denoted by e, s, and h, then, we can calculate r by using trigonometric properties:

$$\cos\theta = \frac{s}{e} = \frac{e}{2r}. \tag{3.2}$$

According to (3.2),

$$r = \frac{e^2}{2s}. \tag{3.3}$$

If we substitute e (3.1) in (3.3), we obtain

$$r = \frac{s^2 + h^2}{2s}. \tag{3.4}$$

In our case, we have $h = \dfrac{C_h(b)}{2}$. If we substitute h in (3.4), we obtain the expression relating $r(b)$ and the c-scale value $C_h(b)$, which is

$$r(b) = \frac{4s^2 + C_h(b)^2}{8s}. \tag{3.5}$$

3.1.2 Relation Between c-scale Value $C_h(b)$ and Arc Length $A(b)$

The perimeter of a circle is given by $P = 2\pi r$. Therefore, the length of a circular arc $A(b)$ with an angle 2α at the centre (Fig. 3.3) is given by

$$A(b) = \frac{2\pi r(b)}{360} \cdot (2\alpha) = \frac{\pi\alpha r(b)}{90}. \tag{3.6}$$

From trigonometric properties of triangles we have

$$\tan\alpha = \frac{h}{r(b) - s} = \frac{C_h(b)}{2(r(b) - s)}. \tag{3.7}$$

Therefore,

$$\alpha = \arctan\left(\frac{C_h(b)}{2(r(b) - s)}\right), \tag{3.8}$$

and the relation between the chord $C_h(b)$ and the arc length $A(b)$ is

$$A(b) = \frac{\arctan\left(\frac{C_h(b)}{2(r(b)-s)}\right)\pi r(b)}{90}. \tag{3.9}$$

3.1.3 Relation Between c-scale Value $C_h(b)$ and Curvature $\kappa(b)$

The curvature κ of a circular segment of radius r is $\kappa = \frac{1}{r}$. Therefore, the relationship between curvature and c-scale value is:

$$\kappa(b) = \frac{8s}{4s^2 + C_h(b)^2}, \tag{3.10}$$

where $C_h(b)$ represents the length of the chord of the osculating circle matching the c-scale segment $C(b)$ at b and s is the shortest distance from the mid-point of the arc to the chord.

3.1.4 Orientation

The c-scale concept can also capture the notion of orientation of the shape boundary at any point b on the boundary. In this section, we introduce the concept of orientation and its calculation. Orientation is an additional feature that is useful in shape description and characterization:

Definition 3.1.3 (Orientation)**.** *Let b_1 be the starting point on a shape boundary $\mathcal{B}$, let b_2 be any other point on $\mathcal{B}$. Let t_1 and t_2 be the tangents at b_1 and b_2, respectively, to $\mathcal{B}$. Then, the* ***orientation*** *$O(b)$ of $\mathcal{B}$ at b_2 is defined as the net total angle, in degrees, t_1 will have to be rotated as one traverses along $\mathcal{B}$ from b_1 to reach b_2, while matching t_1 with the tangent t of each point b along the traversed path.*

This definition is illustrated in Fig. 3.4. In some shapes, $O(b)$ may go beyond 360° for some points as illustrated in Figs. 3.7 and 3.8. In the rest of the section, we will derive an expression for $O(b)$ and a method to easily compute $O(b)$ from the c-scale description of $\mathcal{B}$.

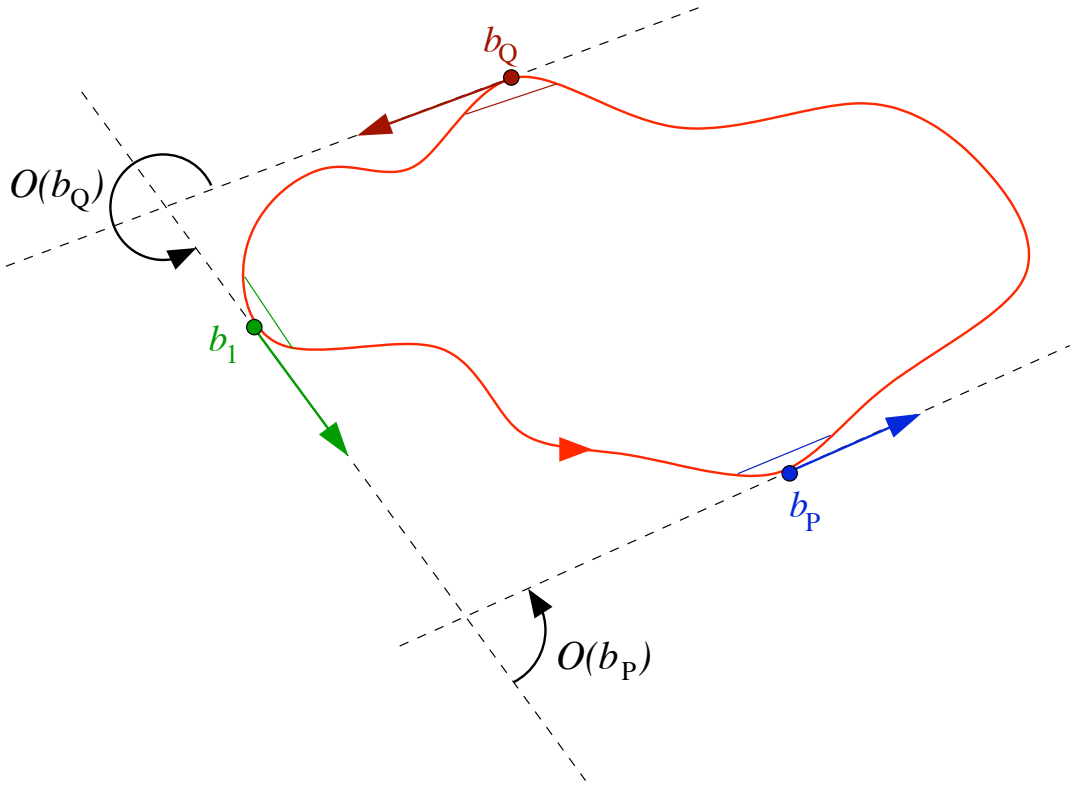


Figure 3.4: Orientation $O(b)$ at any point b of a boundary. The orientation at b_P (and b_Q) is measured with respect to the tangent at the reference starting point b_1.

In the digital setting, we assume the tangent at b to be a line at b parallel to the chord connecting the end points b_b and b_f (Fig. 3.1) of the c-scale segment $C(b)$ at b. The boundary is followed anticlockwise, the inside of the object kept always on the left of the boundary at every bel b (Fig. 3.4). By using the chords found for each bel b, we calculate the unit vector $\mathbf{u}$ of the chords in the direction of the boundary at b.

At any bel b, the unit vector $\mathbf{u}$ of the chord associated with b is defined as

$$\mathbf{u} = \frac{\overrightarrow{b_b b_f}}{\|\overrightarrow{b_b b_f}\|}, \tag{3.11}$$

where $\overrightarrow{b_b b_f}$ is a vector going from the backward bel to the forward bel in $C(b)$, the backward and forward directions being defined with respect to the assumed counter clockwise orienta-

tion of the boundary. We can express b_b and b_f with respect to their coordinates in the x- and y-axis as $b_b = (x_{b_b}, y_{b_b})$ and $b_f = (x_{b_f}, y_{b_f})$, and

$$\|\overrightarrow{b_b b_f}\| = \sqrt{(x_{b_f} - x_{b_b})^2 + (y_{b_f} - y_{b_b})^2} = C_h(b). \tag{3.12}$$

Substituting (3.12) in (3.11), we can define the components of $\mathbf{u}$ with respect to the x- and y-axis as

$$\mathbf{u} = \frac{\overrightarrow{b_b b_f}}{C_h(b)} = \left(\frac{x_{b_f} - x_{b_b}}{C_h(b)}, \frac{y_{b_f} - y_{b_b}}{C_h(b)} \right). \tag{3.13}$$

Each unit vector $\mathbf{u}$ has two components: the direction cosines u_x, u_y, as represented in Figs. 3.5 and 3.6. The direction cosines characterize the angles that the unit vector makes

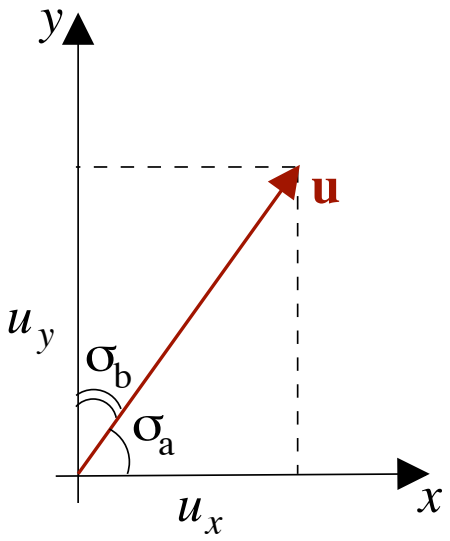


Figure 3.5: Components of a unit vector $\mathbf{u}$. $\mathbf{u}$ is at an angle σ_a with respect to the x-axis, and at an angle σ_b with respect to the y-axis. u_x and u_y are the components of $\mathbf{u}$ along the x- and y-axis, respectively.

with the positive x- and y-axis, respectively. In Fig. 3.5, the angle between the unit vector $\mathbf{u}$ and the x-axis is σ_a, whereas, the angle between $\mathbf{u}$ and the y-axis is σ_b. Therefore, since the magnitude of $\mathbf{u}$ is equal to 1, the direction cosines u_x and u_y are defined as

$$u_x = \cos \sigma_a, \tag{3.14}$$

and

$$u_y = \sqrt{1 - u_x^2} = \cos \sigma_b, \tag{3.15}$$

with

$$\cos^2 \sigma_a + \cos^2 \sigma_b = 1. \tag{3.16}$$

To characterize the whole range between 0 and 360 degrees via direction cosines, we need to use both components u_x and u_y, as u_x can distinguish angles only in the range $0° - 180°$. Therefore, we use the sign of u_y to cover the whole 360° range, making it possible to distinguish among the quadrants, as shown in Fig. 3.6.

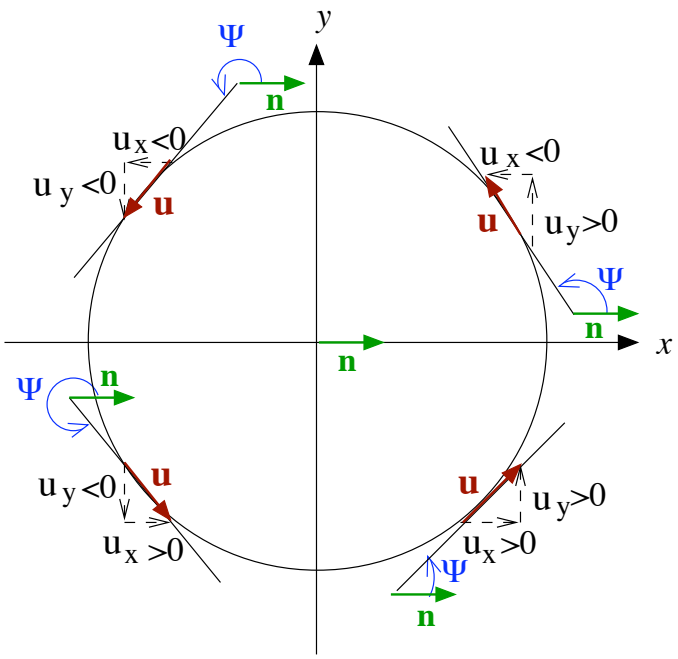


Figure 3.6: Orientation using direction cosines.

From the direction cosines, we can write down the tangent orientation at b as

$$\Psi(b) = \begin{cases} 360 - \arccos(u_x(b)), & \text{if } u_y < 0 \\ \arccos(u_x(b)), & \text{if } u_y \geqslant 0. \end{cases} \tag{3.17}$$

We can relate tangent orientation $\Psi(b)$ at b to the c-scale value $C_h(b)$ by substituting the components of $\mathbf{u}$ (3.13) in (3.17) as

$$\Psi(b) = \begin{cases} 360 - \arccos\left(\dfrac{x_{b_f} - x_{b_b}}{C_h(b)}\right), & \text{if } \left(\dfrac{y_{b_f} - y_{b_b}}{C_h(b)}\right) < 0 \\ \arccos\left(\dfrac{x_{b_f} - x_{b_b}}{C_h(b)}\right), & \text{if } \left(\dfrac{y_{b_f} - y_{b_b}}{C_h(b)}\right) \geqslant 0. \end{cases} \tag{3.18}$$

Since $C_h(b)$ is always positive, we can simplify (3.18) as

$$\Psi(b) = \begin{cases} 360 - \arccos\left(\dfrac{x_{b_f} - x_{b_b}}{C_h(b)}\right), & \text{if } y_{b_f} < y_{b_b} \\ \arccos\left(\dfrac{x_{b_f} - x_{b_b}}{C_h(b)}\right), & \text{if } y_{b_f} \geqslant y_{b_b}. \end{cases} \tag{3.19}$$

Note that $\Psi(b)$ varies between 0° and 360°. For a variety of reasons, it is better to measure the orientation $O(b)$ of the shape at any bel b with respect to the orientation at the starting point b_1. Our goal is to derive $O(b)$ from $\Psi(b)$ by using a cumulative procedure as described below.

By definition, we set $O(b_1) = 0^\circ$. To have orientation of smooth boundaries change continuously along the boundary and without creating artificial discontinuities that will be introduced if orientation were to be expressed in a modulo 360 format, we express orientation to be any number ($-ve$ or $+ve$) on the real line. Thus, in a shape, like the one in Fig. 3.7, depending on the number of anti-clockwise or clockwise windings (spirals) the shape makes, the orientation $O(b)$ may be greater than 360° or less than -360° depending on how many turns the chord at b_1 has undergone in reaching b starting from b_1. These ideas are captured in the following recursive definition of $O(b)$.

$$O(b_1) = 0^\circ, \tag{3.20}$$

and for $i > 1$,

$$O(b_i) = O(b_{i-1}) + \mathit{diff}(b_i), \tag{3.21}$$

where, with $\triangle(b_i) = \Psi(b_i) - \Psi(b_{i-1})$,

$$diff(b_i) = \begin{cases} \triangle(b_i) - 360, & \text{if } \triangle(b_i) > 180^\circ \\ \triangle(b_i) + 360, & \text{if } \triangle(b_i) < -180^\circ \\ \triangle(b_i), & \text{otherwise} \end{cases} \tag{3.22}$$

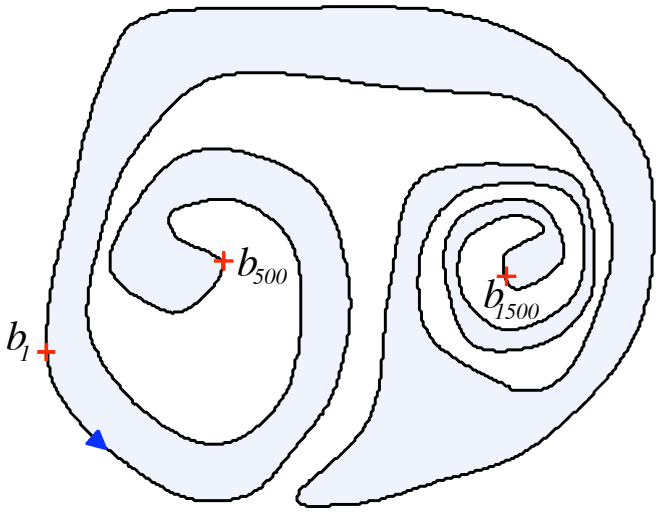


Figure 3.7: Oriented boundary of a spiral shape, with the reference point b_1 and the assumed orientation denoted by the blue arrow.

We note that, in the above description, $O(b)$ may be any number on the real line. This way we can describe shapes, such as spirals (Fig. 3.7), that have several turns, and represent the real angle at each point of the boundary, which may go beyond 360° and below −360°, as shown in Fig. 3.8 for the boundary of Fig. 3.7. Note that this strategy also considers the direction in which the turns are made in the boundary with respect to its assumed orientation. Thus, for the spiral on the left, $O(b)$ increases up to the point close to b_{500} accounting for continual anti-clockwise rotation of the tangent vector. Subsequently, clockwise rotation of the boundary ensues up till the point close to b_{1500} causing $O(b)$ to decrease. O(b) then increases, signalling anti-clockwise rotations of the tangent until b_1 is reached.

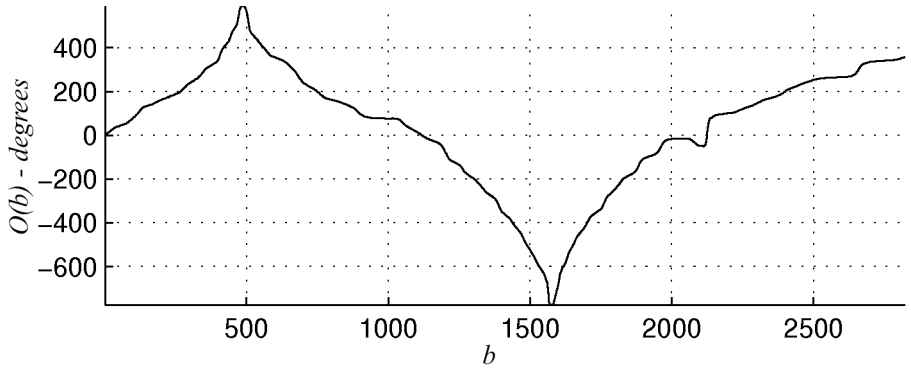


Figure 3.8: Orientation $O(b)$ for the boundary in Fig. 3.7, with the reference point b_1 as indicated.

3.1.5 Relation Between Orientation $O(b)$ and Curvature

Curvature is usually defined as the rate of change of the slope of the tangent at each point on the boundary. Therefore, the first derivative $O'(b)$ of orientation $O(b)$ yields curvature information. By using $O'(b)$, we note that information about the orientation of the boundary is also captured in the obtained curvature, and therefore, we have more information than using only the magnitude of the curvature. By using this approach we can locate special points of interest on $\mathcal{B}$. Local positive maxima of $O'(b)$ will correspond to convex corners, local negative minima to concave corners, constant zero curvature to straight lines, constant non-zero curvature to circular segments, and zero crossings to inflection points, etc. In this manner, we have a complete and comprehensive description of the boundary by using $O'(b)$.

3.1.6 Example

The above theory is illustrated by using an example shape in Fig. 3.9. This shape is constructed from theoretical functions. It includes different parts such as a rotated rectangle, circular arcs of different radii (convex and concave), and a sine wave. The boundary is oriented and follows the direction of the arrow, starting from the reference point b_1, leaving always the inside of the object to the left. The order of the bels is defined using this direction, and the coordinates of the bels are computed by using the functions that define the different sections.

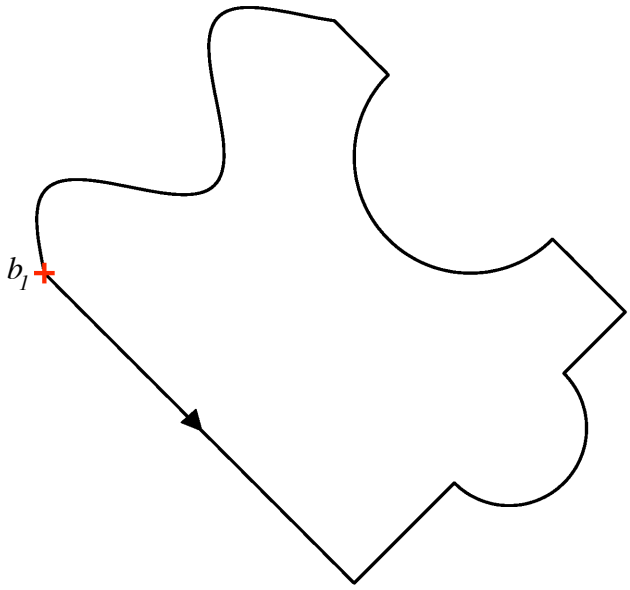


Figure 3.9: A general shape formed by straight lines, circular arcs, and a sine wave.

To this shape, we apply the c-scale calculation method with $t = 0.02$. This parameter controls the level of detail. For digital boundaries, we usually set $t \approx 3$, which works well for all shapes we tested. This value of t is able to preserve appropriate boundary details and at the same time ward against digitization noise. The values of the chord lengths (i.e., c-scale) at each bel are represented in Fig.3.10.

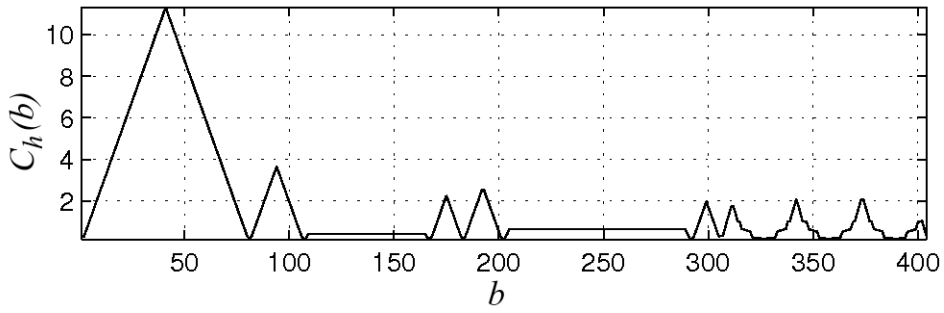


Figure 3.10: The c-scale value $C_h(b)$ for the shape in Fig. 3.9 starting with b_1 at origin.

We observe that large $C_h(b)$ represents small curvature at b, whereas small $C_h(b)$ indicates

high curvature at b. Due to the symmetry of the c-scale method, each straight segment on the boundary corresponds to a peak in the $C_h(b)$ plot, and the location of the peak corresponds to the midpoint of the segment. Chords of same length are obtained when we have a circular shape with constant curvature. And valleys in the $C_h(b)$ plot represent curved regions of the boundary. By using Equation (3.9), a representation of the arc length $A(b)$ (as shown in Fig. 3.11) can be made. In this case, $A(b)$ and $C_h(b)$ are very similar, but this may not

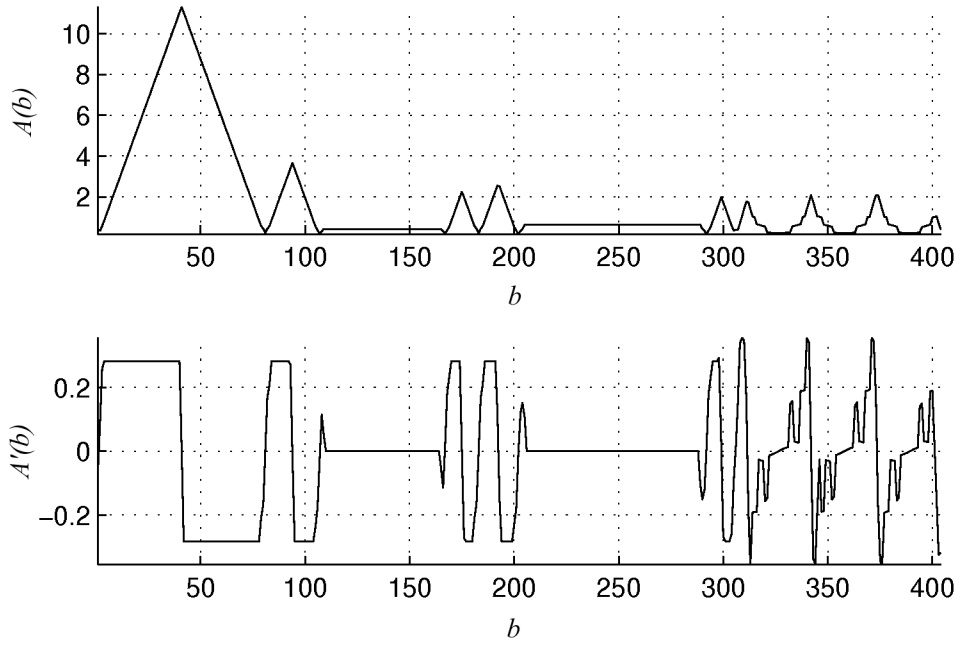


Figure 3.11: Arc length $A(b)$ and its derivative $A'(b)$ for the shape in Fig. 3.9 starting with b_1 at origin.

be the case for digital boundaries. The peaks in $A(b)$ correspond to middle points of flat segments or inflection points. The valleys in $A(b)$ represent high curvature points or corners. By detecting these peaks and valleys we will be able to detect the characteristic points of the shape. We can use peaks, or valleys, or both for describing shape.

The orientation $O(b)$, as well as its derivative $O'(b)$, for each bel for the shape in Fig. 3.9 are shown in Fig. 3.12. $O'(b)$ will be very useful for shape description as it constitutes a good estimation of curvature, allowing additionally to distinguish between convex and concave

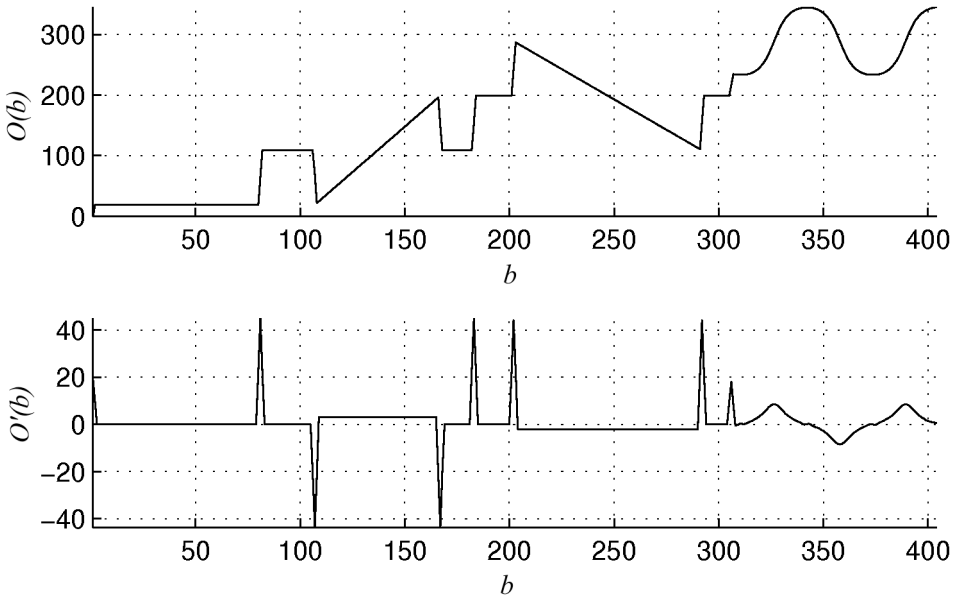


Figure 3.12: Orientation $O(b)$ and its derivative $O'(b)$ for the shape in Fig. 3.9 starting with b_1 at origin.

regions in the boundary.

According to Equation (3.10), we represent the curvature of the shape under consideration in Fig.3.13. Straight lines on the boundary have a curvature equal to zero, circular regions

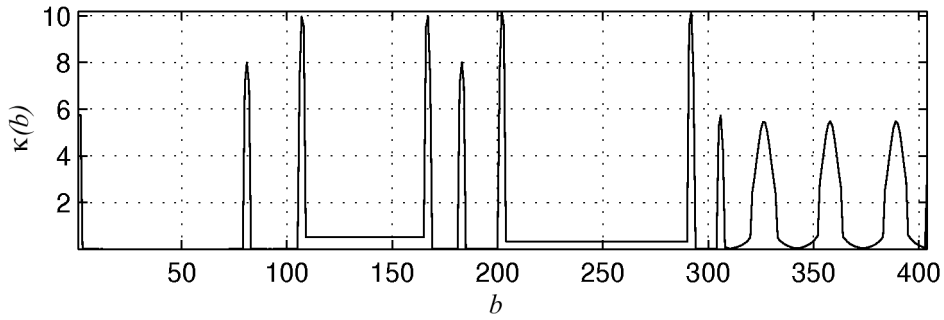


Figure 3.13: Curvature $\kappa(b)$ for the shape in Fig. 3.9 starting with b_1 at origin.

have constant curvature and sine waves and corners have high curvature values.

3.2 Algorithms

The c-scale algorithm is summarized below.

Algorithm 3.1: c-**scale**

Input: Boundary $\mathcal{B} = \{b_i : i = 1, ..., q\}$ and scale parameter t.

Output: c-scale value $C_h(b_i)$ and c-scale segment $C(b_i)$, $i = 1, ..., q$.

for *each* $b_i, i = 1 : q$ **do**

1. Set $b_f = b_{i+1}$ and $b_b = b_{i-1}$;
2. Set $C(b_i) = [b_b : b_f]$ and $d = 0$;

while $d < t$ **do**

3. Find the maximum distance d for the bels in $C(b_i)$ to the line $b_b b_f$;
4. Set $b_f = b_{f+1}$ and $b_b = b_{b-1}$;
5. Update $C(b_i) = [b_b : b_f]$;

end

6. $b_f = b_{f-1}$ and $b_b = b_{b+1}$;
7. Update $C(b_i) = [b_b : b_f]$;
8. Calculate $C_h(b_i)$ as the length of the segment $\overline{b_b b_f}$;

end

10. Output $C_h(b_i)$ and $C(b_i)$ for all i.

By using the c-scale algorithm, the orientation at each bel b in a boundary $\mathcal{B}$ is obtained as per the following algorithm.

Algorithm 3.2: Orientation

Input: Boundary $\mathcal{B} = \{b_i : i = 1, ..., q\}$, c-scale values $\{C_h(b_i) : i = 1, ..., q\}$, and c-scale segments $\{C(b_i) : i = 1, ..., q\}$.

Output: c-scale orientation $O(b_i)$, $i = 1, ..., q$.

Auxiliary Data: Tangent orientations $\Psi(b_i), i = 1, ..., q$.

1. Set $O(b_1) = 0$;

for *each* $b_i, i = 1 : q$ **do**

2. Using $C(b_i)$, find b_f and b_b associated with that segment;
3. Compute $u_x(b_i) = \dfrac{(x_f - x_b)}{C_h(b_i)}$ and $u_y(b_i) = \dfrac{(y_f - y_b)}{C_h(b_i)}$;
4. **if** $u_y(b_i) < 0$, **then** $\Psi(b_i) = 360 - \arccos u_x(b_i)$;
   else $\Psi(b_i) = \arccos u_x(b_i)$;

end

for *each* $b_i, i = 2 : q$ **do**

5. Set $\triangle(b_i) = \Psi(b_i) - \Psi(b_{i-1})$;
6. **if** $\triangle(b_i) > 180$, **then** set $\mathit{diff}(b_i) = \triangle(b_i) - 360$;
7. **if** $\triangle(b_i) < -180$, **then** set $\mathit{diff}(b_i) = \triangle(b_i) + 360$;
8. **else** set $\mathit{diff}(b_i) = \triangle(b_i)$;
9. Set $O(b_i) = O(b_{i-1}) + \mathit{diff}(b_i)$;

end

10. Output $O(b_i)$ for all i;

3.3 A Method for Shape Description Using c-scale

In this section, we present the method of boundary shape description that uses c-scale concepts. Given a (digital) boundary $\mathcal{B}$ and a scale parameter t, our goal is to obtain a partition $P_{\mathcal{B}}$ of $\mathcal{B}$ into segments, a set s_L of landmarks (or characteristic dominant points), and a shape description assigned to every element of $P_{\mathcal{B}}$ and s_L. The method is summarized in Fig.3.14.

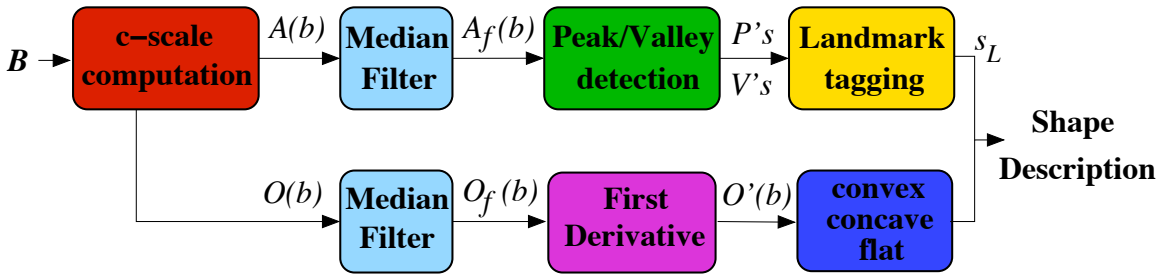


Figure 3.14: The method of boundary shape description.

First, we determine at each point b of $\mathcal{B}$ the c-scale value $C_h(b)$ via Algorithm 3.1 from which we estimate arc length $A(b)$ via Equation 3.9 and orientation $O(b)$ via via Algorithm 3.2. Second, we smooth $A(b)$ and $O(b)$ by using a median filter. After obtaining the smoothed versions $A_f(b)$ and $O_f(b)$, we detect peaks and valleys in $A_f(b)$ by using mathematical morphology operators. The peaks correspond to the middle points of straight line segments in the boundary and the valleys to the middle points of curved segments. Then, by using $O_f'(b)$, we can assign a definition to each dominant point, such as convex, concave, or flat. This way, we obtain a complete description of shape in terms of dominant points or landmarks. Landmarks can be selected at peaks, valleys, or both. In the following, we explain each step of the process in detail.

3.3.1 Median Filter

The median filter is a non-linear digital filter used to smooth the signal, retaining its general shape, while reducing the noise. For each element b, and for a particular window size used, it assigns the median value of the elements in the window to b. In our case, we smooth $A(b)$ and $O(b)$ by using a median filter of width $2w+1$ centred at every element b, where w is

the half width of the window used for filtering, specified in terms of the number of points considered on either side of b. We repeat this process m times on the entire boundary to get a smoothed version of $A(b)$ and $O(b)$, called $A_f(b)$ and $O_f(b)$, respectively. This way, we reduce the noise of the digital signals and avoid detecting spurious peaks or valleys. The median filtering process on $A(b)$ is illustrated on a rabbit shape in Fig. 3.15. We used a window size of $w = 2$, and repeated the process $m = 2$ times to smooth $A(b)$ (Fig. 3.16(a)). The result, $A_f(b)$, is presented in Fig. 3.16(b).

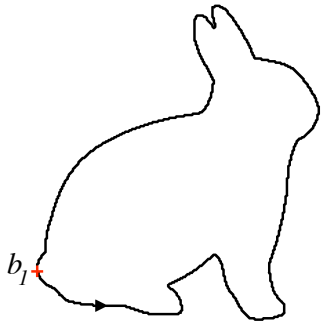


Figure 3.15: A digital boundary of a rabbit shape. b_1 is the starting point and the boundary is followed in the direction of the arrow.

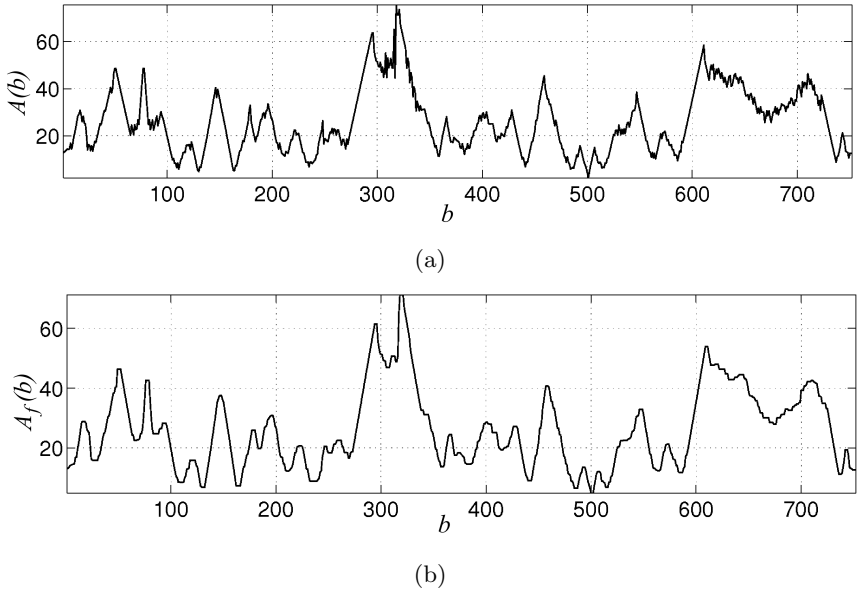
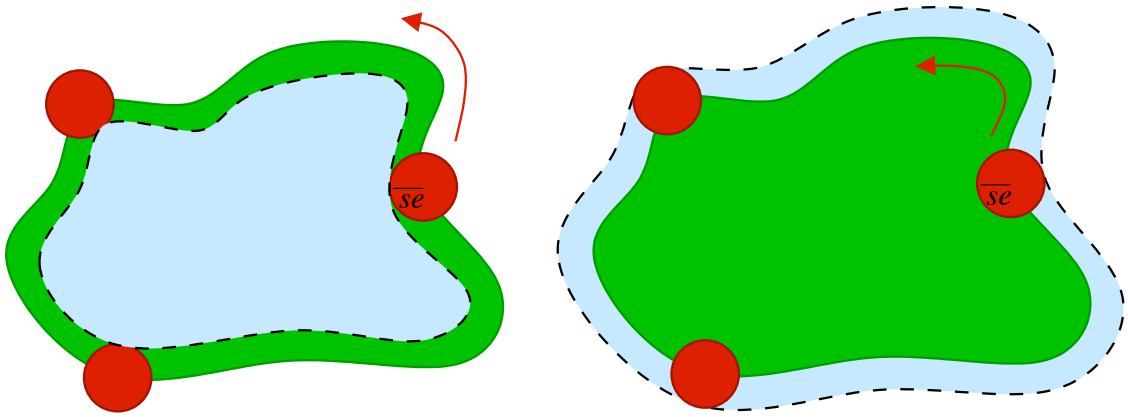


Figure 3.16: Median filtering with $w = 2$ and $m = 2$ for the rabbit shape. (a)$A(b)$. (b)$A_f(b)$.

3.3.2 Peak and Valley Detection Using Mathematical Morphology

The peaks in $A_f(b)$ correspond to straight line segments or inflection points in the boundary, whereas the valleys in $A_f(b)$ concur with curved segments in $\mathcal{B}$. We automatically detect peaks and valleys at different levels of detail by using mathematical morphology operators.

Mathematical morphology [Serra, 1982][Sonka et al., 1999] is based on set theory and provides powerful tools for image analysis. Fundamental operations are erosion, dilation, opening and closing. A structuring element defines the size and shape of the transformation to be done, as shown in Fig. 3.17. An erosion of an object by a structuring element (Fig. 3.17(a)) shrinks the original object with respect to the shape and size of the structuring element used, whereas a dilation (Fig. 3.17(b)) expands the object.



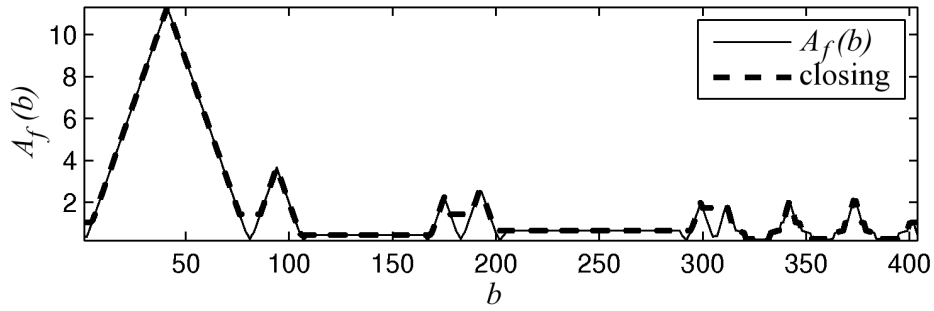
(a) Erosion. (b) Dilation.

Figure 3.17: An object (in green) and its morphological (a) erosion and (b) dilation by a circular structuring element of size se. Resulting objects are shown in pale blue.

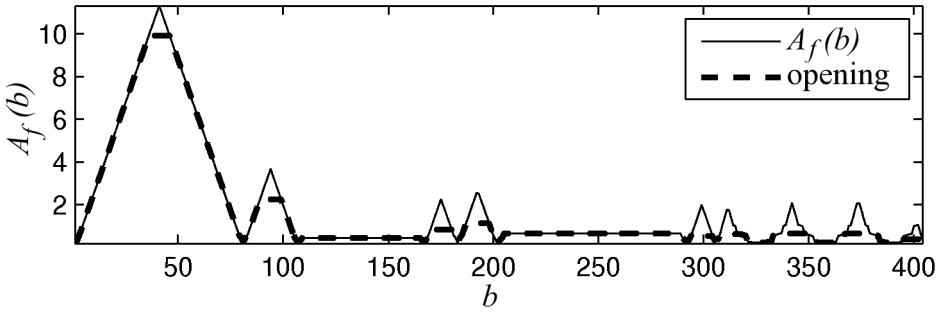
These operations can be defined for binary and grey scale images, as well as for 1D signals. In our case, we use a structuring element of size se applied to the signal $A_f(b)$. To perform an erosion on $A_f(b)$, we need to replace each element's value by the minimum value within a window of size $2se+1$. Similarly, to dilate $A_f(b)$, we assign to each element the maximum value within the window centred at that element. Note that these assignments are made not in a sequential manner as the window is moved but in a parallel manner at the end after all

positions are examined.

Using the concepts of erosion and dilation, we can now define the operations of closing and opening. A closing is defined as a dilation followed by an erosion. This transformation preserves the extrusions removing the intrusions. When operated on a 1D signal, it fills the valleys while leaving the peaks unchanged (Fig. 3.18.a). An opening consists of an erosion followed by a dilation. It removes small extrusions on the object, while leaving the intrusions unaltered (Fig. 3.18.b). When applied to a 1D signal, this operation removes the peaks and preserves the valleys. Opening and closing are the transformations we need to detect the peaks and the valleys of $A_f(b)$, as shown in Fig. 3.18 for the shape in Fig. 3.9.



(a) Closing.



(b) Opening.

Figure 3.18: Closing and opening applied on $A_f(b)$ for the shape in Fig. 3.9. (a) Valley detection. (b) Peak detection.

In particular, to find the valleys, we apply to $A_f(b)$ a bottom-hat filtering operation, called BH, which finds the difference between $A_f(b)$ and its closing, defined as

$$BH(A_f(b)) = A_f(b) - Closing(A_f(b)). \tag{3.23}$$

This filter extracts only the valleys of $A_f(b)$ (Fig. 3.19). Once we have detected all the valleys

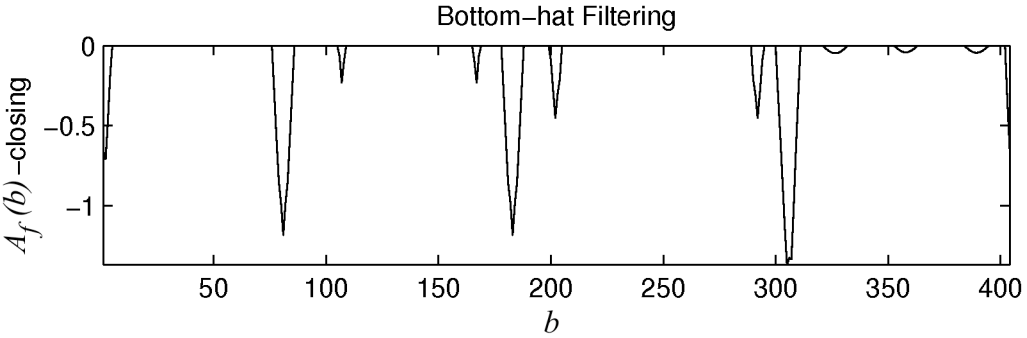


Figure 3.19: Valley extraction using bottom-hat filtering on $A_f(b)$ for the shape in Fig. 3.9.

in $A_f(b)$ we find the minimum value for each valley detected. These local minima correspond to the valley locations in $A_f(b)$ and represent points with high curvature in $\mathcal{B}$. Similarly, to find the peaks, we need a top-hat filtering operation, called TH, which finds the difference between $A_f(b)$ and its opening, defined as

$$TH(A_f(b)) = A_f(b) - Opening(A_f(b)). \tag{3.24}$$

This filter extracts only the peaks of $A_f(b)$ (Fig. 3.20).

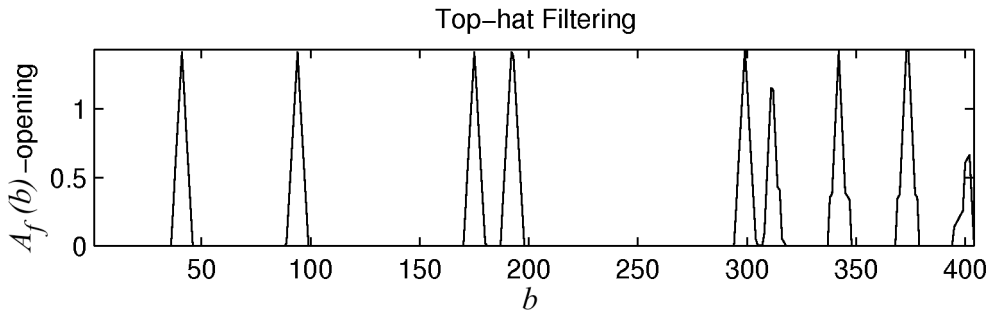
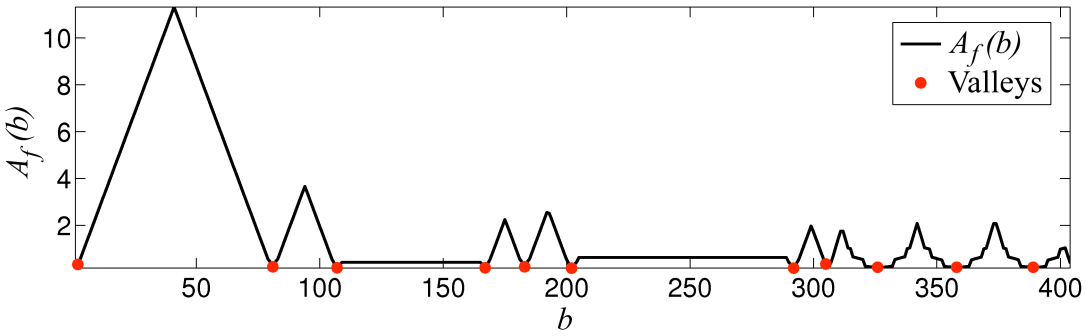


Figure 3.20: Peak extraction using top-hat filtering on $A_f(b)$ for the shape in Fig. 3.9.

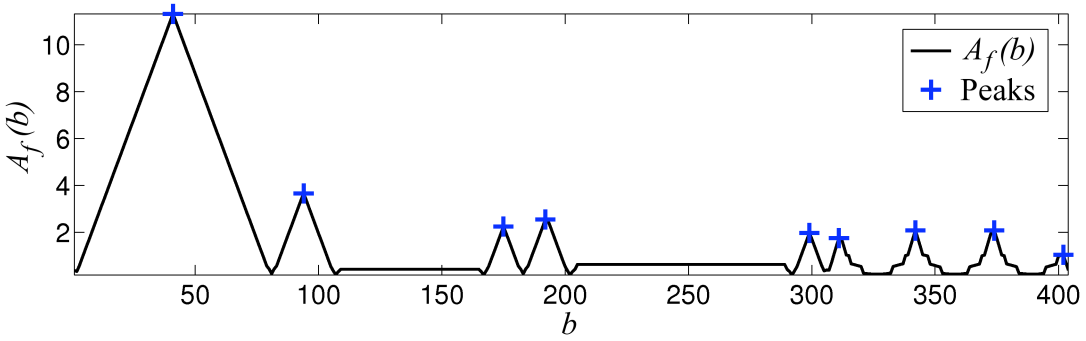
Once we have detected all the peaks in $A_f(b)$ we find the maximum value for each peak detected. These local maxima correspond to the peak locations in $A_f(b)$, which are the middle points of straight segments or inflection points in $\mathcal{B}$.

By selecting a different size (se_v and se_p) for the structuring elements, we can vary the number of valleys and peaks detected. Let θ_p be the parameter controlling the size of the detected peaks, and θ_v the parameter controlling the size of the detected valleys. We can avoid spurious valleys and peaks, or select most prominent ones, by keeping only those peaks and valleys that are greater than a certain value. This allows us to fully control the number of dominant points we want for a given application.

For the example in Fig. 3.9, we used a structuring element of size 5 for both valley and peak detection. No element selection was necessary in this case by specifying θ_v and θ_p. After the bottom-hat and the top-hat filtering operations, we detect the peaks and valleys perfectly as shown in Fig. 3.21.



(a) Valley detection.

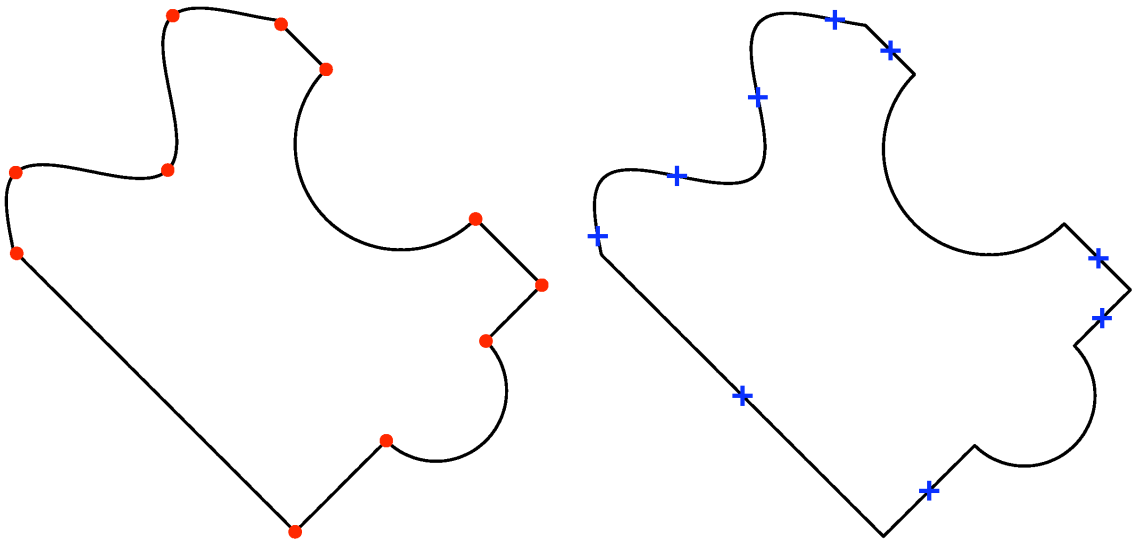


(b) Peak detection.

Figure 3.21: Detection of valleys and peaks on $A_f(b)$ for the shape in Fig. 3.9.

Once we have detected the peaks P and the valleys V in $A_f(b)$, we can locate the land-

marks s_L on the boundary as shown in Fig. 3.22 by simply identifying the points where local minima/maxima occurred.



(a) (b)

Figure 3.22: Dominant points detected on the shape in Fig. 3.9. (a) High curvature points. (b) Middle points of straight line segments or inflection points.

Another way of selecting the level of detail we want to consider is by using two threshold values nP and nV. nP will be the smallest value in $A_f(b)$ from which we want to consider peaks. nV will be the highest value in $A_f(b)$ from which we want to consider valleys. This is used in the case when we want to consider only more prominent valleys (corresponding to higher curvature regions) and more prominent peaks (corresponding to lower curvature regions).

3.3.3 Complete Shape Description

By using $O_f^{'}(b)$ (Fig. 3.12), we can attain a complete shape description of $\mathcal{B}$ and at each bel in $\mathcal{B}$. Recall that local positive maxima correspond to convex corners, local negative minima to concave corners, constant zero curvature to straight line segments, constant non-zero curvature to circular segments, and zero crossings to inflection points. In this manner,

we can assign a shape description to every element of $P_{\mathcal{B}}$ and s_L by using $O_f^{'}(b)$. This can be achieved by decomposing the shape into pieces such as: corners, straight lines, and circular segments. Then, we associate with each piece the corresponding description by using $O_f^{'}(b)$. Clearly, it is also possible to assign a description to each bel b of $\mathcal{B}$ by noting the segment to which b belongs.

Corner Characterization

Corners or high curvature points, are detected in $\mathcal{B}$ as described in 3.3.2 (Fig. 3.23). Subsequently,

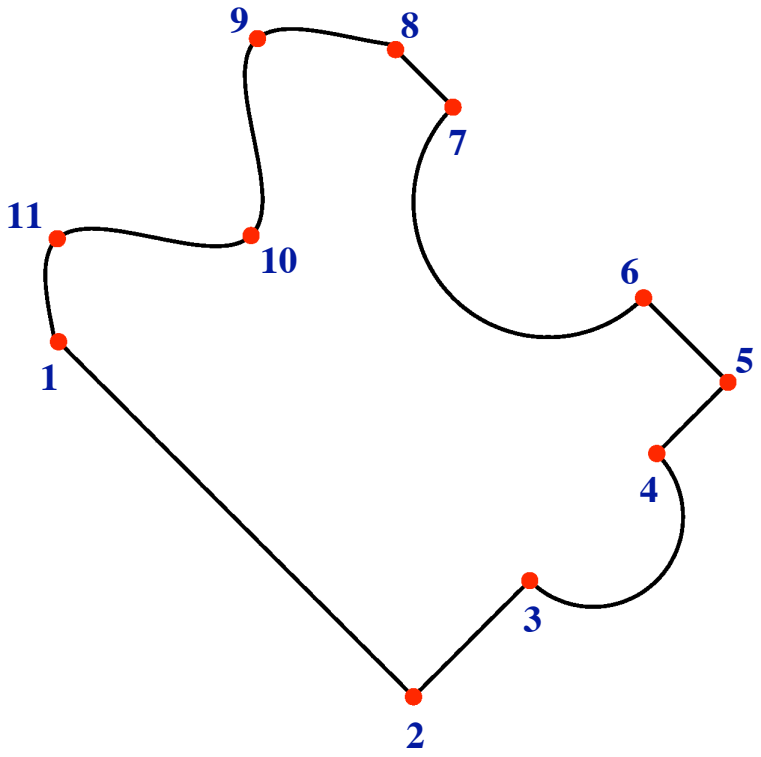


Figure 3.23: Numbered high curvature points on the shape in Fig. 3.9.

we find their corresponding positions in $O_f^{'}(b)$, as shown in Fig. 3.24. (Although for the mathematical shape of Fig. 3.9, no filtering is needed and we can use $O'(b)$ directly, for digital representations of natural shapes, we should use $O_f^{'}(b)$ - the derivative of $O_f(b)$.) As we can see, corners correspond to local maxima and local minima in $O_f^{'}(b)$. Once identified, we assign a description to each such point by using the sign of $O_f^{'}(b)$ at these points. If

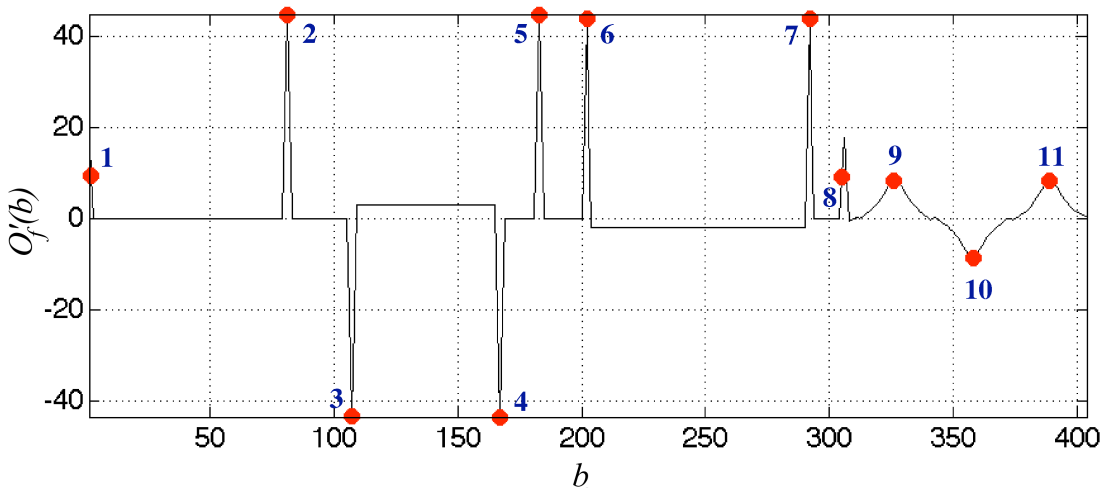


Figure 3.24: Corresponding high curvature points in $O'_f(b)$ for the shape in Fig. 3.9.

Table 3.1: Description of high curvature points by using the sign of $O'_f(b)$.

Landmark	$O'_f(b)$ **sign**	**Description**
1	+	convex
2	+	convex
3	−	concave
4	−	concave
5	+	convex
6	+	convex
7	+	convex
8	+	convex
9	+	convex
10	−	concave
11	+	convex

$O'_f(b) > 0$, then the corner is convex; if $O'_f(b) < 0$, it is concave. Table 3.1 gives the associated description for each corner detected in Fig. 3.9 with respect to its sign in $O'_f(b)$. This information is potentially useful for certain applications in the shape analysis field as it adds a feature to the corners apart from just their location.

Straight Line Segment Extraction and Characterization

Middle points of straight segments, or inflection points, are detected in $\mathcal{B}$ as described in 3.3.2 (Fig. 3.25). Subsequently, we find their corresponding locations in $O_f^{'}(b)$, as shown in

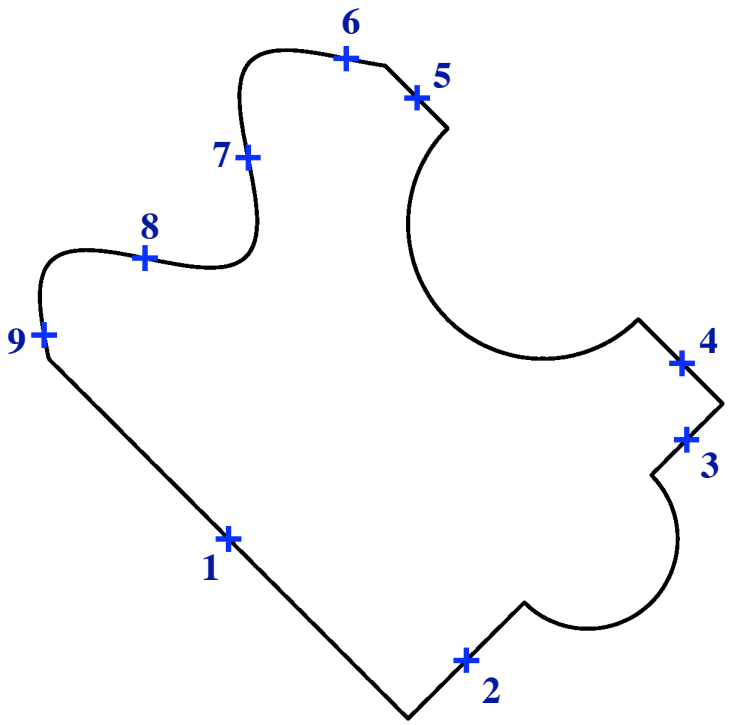


Figure 3.25: Numbered inflection points on the shape in Fig. 3.9.

Fig. 3.26. We observe that these points are located in regions where $O_f^{'}(b) = 0$. By identifying all those connected regions where $O_f^{'}(b) = 0$, and those that are associated with each inflection point, we can automatically determine the straight segments in $\mathcal{B}$. The connected regions identified in $O_f^{'}(b)$, and that correspond to the straight segments in $\mathcal{B}$, are represented in Fig. 3.27 for the shape in Fig. 3.9.

In Fig. 3.28, we display the detected straight segments for the shape in Fig. 3.9 and that correspond to the connected regions in $O_f^{'}(b)$ with $O_f^{'}(b) = 0$ (Fig. 3.27). Each straight segment is represented in a different color to show the correspondence between segments in Fig. 3.27 and Fig. 3.28.

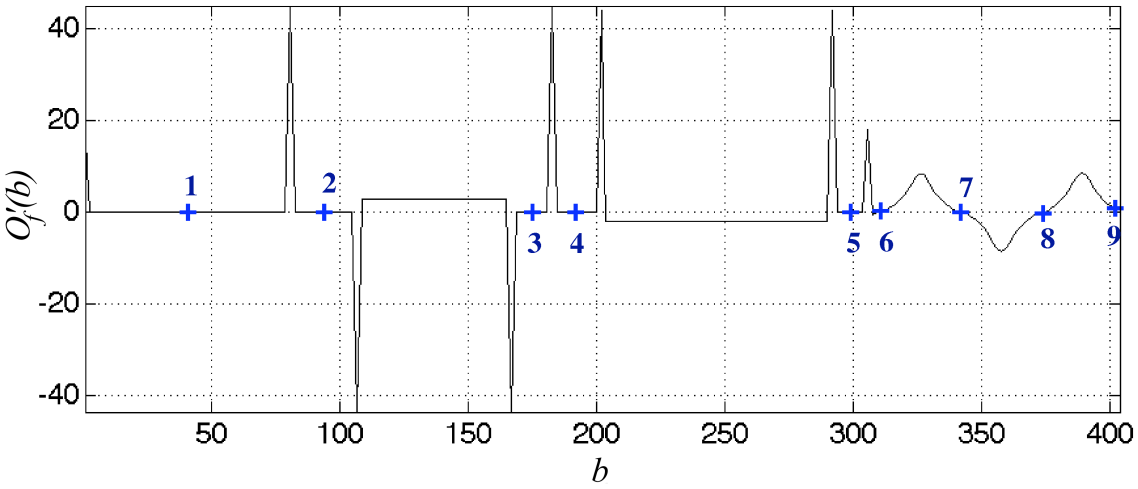


Figure 3.26: Corresponding middle points of straight segments in $O'_f(b)$ for the shape in Fig. 3.9.

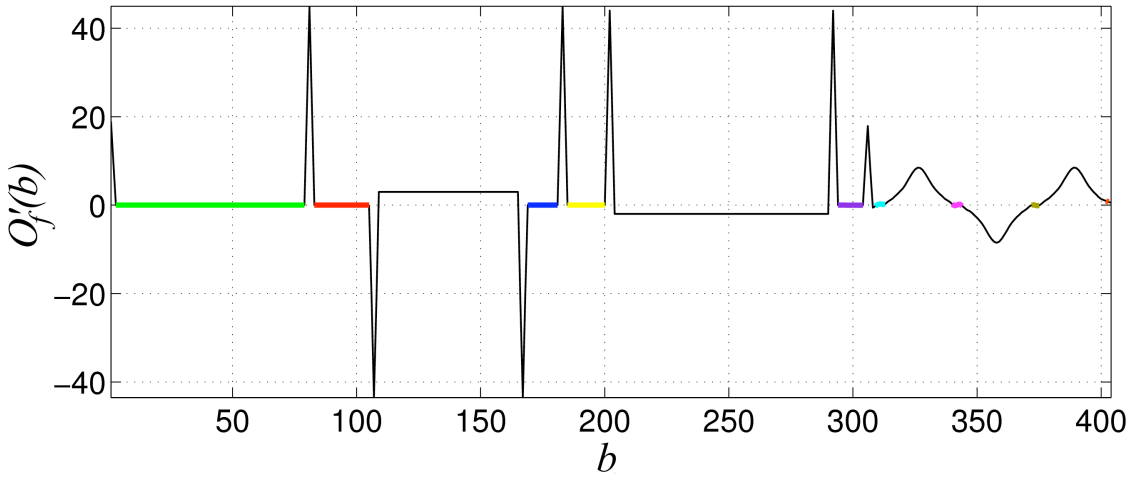


Figure 3.27: Components of connected regions in $O'_f(b)$ with $O'_f(b) = 0$, corresponding to straight segments in $\mathcal{B}$ for the shape in Fig. 3.9 are displayed as different color segments.

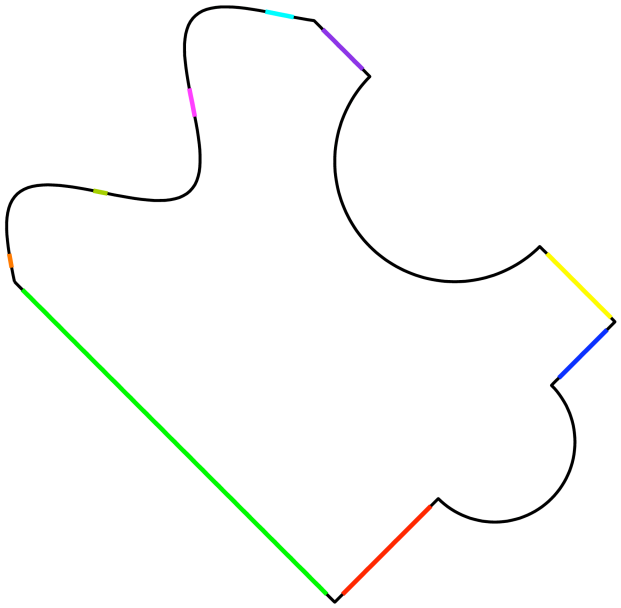
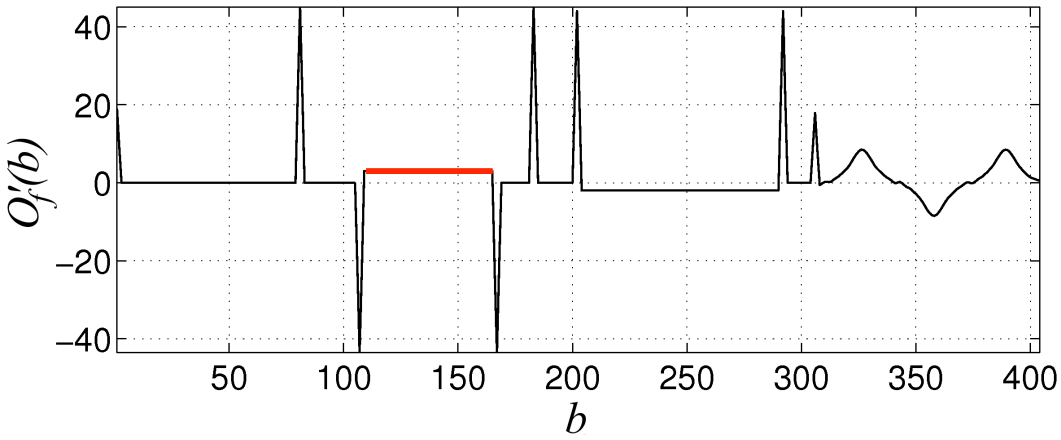


Figure 3.28: Straight segments detected for the shape in Fig. 3.9.

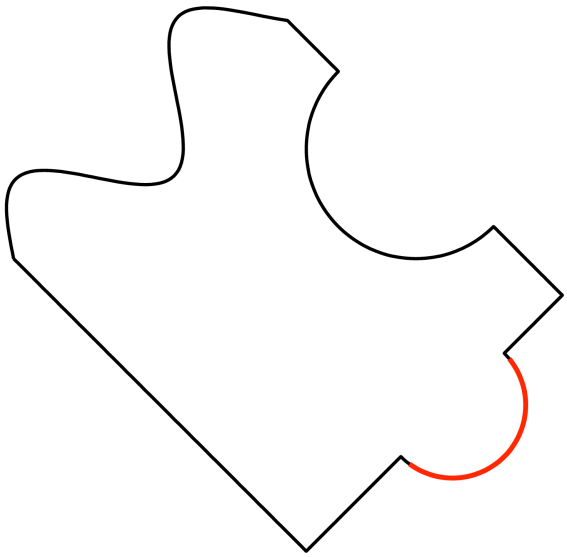
Circular Segment Extraction and Characterization

The last step consists of identifying and describing any circular segments appearing in $\mathcal{B}$. These segments correspond to connected regions in $O_f^{'}(b)$ where $O_f^{'}(b) = constant \neq 0$. After finding these regions, we perform labelling and extract all the connected pieces in $O_f^{'}(b)$ corresponding to the circular segments automatically (Figs. 3.29(a) and 3.30(a)). Depending on the sign of $O_f^{'}(b)$, the circular segments can be identified to be concave or convex. A positive and constant segment in $O_f^{'}(b)$ defines a convex circular segment (Fig. 3.29(b)), whereas a negative and constant segment in $O_f^{'}(b)$ represents a concave circular segment (Fig. 3.30(b)).

The amplitude of $|O_f^{'}(b)|$ at these regions, represents the degree of curvature of these circular segments. The higher the amplitude, the greater is the curvature of the segment. In digital boundaries of real shapes, perfect circular segments may not occur. However, shape segmentations similar to those presented above can be obtained via approximations of the above exact procedures.

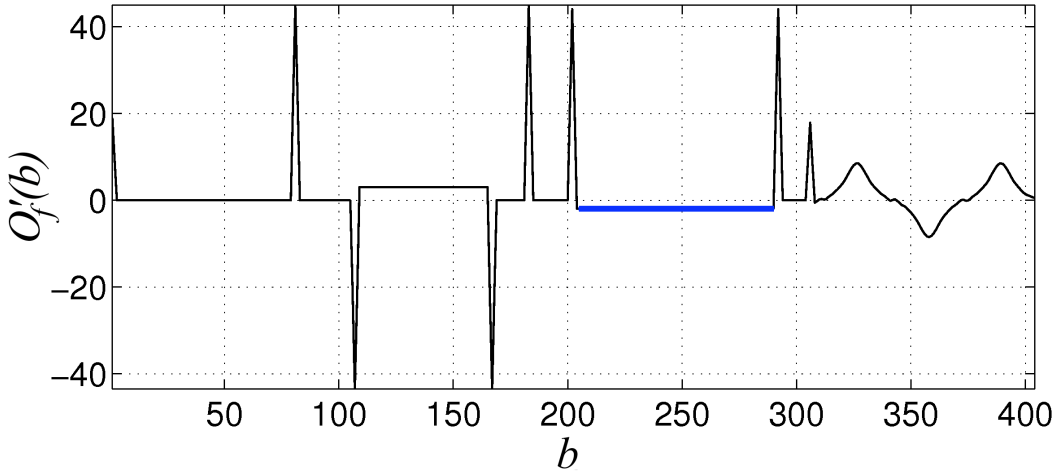


(a)

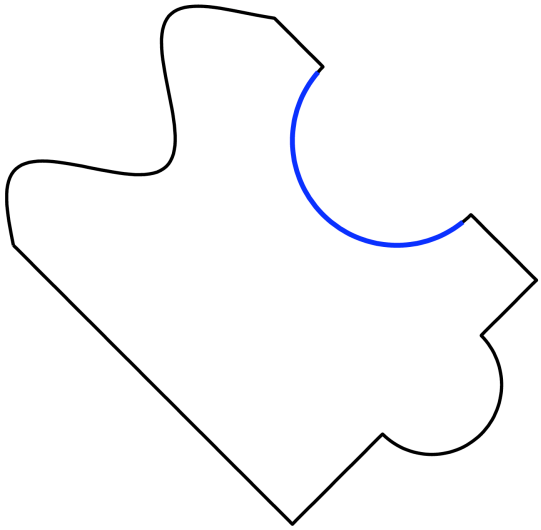


(b)

Figure 3.29: Detection of a convex circular segment at $O'_f(b)$ and $\mathcal{B}$ for the shape in Fig. 3.9. (a) $O'_f(b) = constant > 0$. (b) Convex arc.



(a)



(b)

Figure 3.30: Detection of a concave circular segment on $O'_f(b)$ and $\mathcal{B}$ for the shape in Fig. 3.9. (a) $O'_f(b) = constant < 0$. (b) Concave arc.

In this section, our goal was to explore the full potential of the c-scale shape description principles as they can be used in many different ways depending on the application. Other shape descriptors published in the literature do not provide such a comprehensive characterization of shape.

3.4 Conclusion

We have presented a new theory and method for shape description based on the novel concept of c-scale. The method is simple and produces a comprehensive description of shape with numerous potential applications. For each boundary element b, the arc length of the largest homogeneous curvature region is estimated as well as the orientation of the tangent at b. The method is different from previous methods of curvature estimation and can be directly applied to digital boundaries without requiring prior approximations of the boundary. The shape descriptor performs at different levels of detail by considering the local morphometric scale of the object. In this chapter, we have shown that this method is useful for shape description as well as for the extraction of dominant points.

In the following chapter, we will focus on describing shapes by using only a set of dominant points or landmarks and not the full descriptive means, since we intend to compare the c-scale methods with the state-of-the-art techniques for similar shape characterization tasks. These landmarks may be high curvature points, middle points of straight segments, or both, as defined in this chapter.

4

Evaluation of c-Scale Shape Description Methods

Scale is a fundamental concept in computer vision and pattern recognition, employed especially in the fields of shape analysis, image segmentation, and registration. It represents the level of detail of object information in scenes. Global scale methods in image processing process the scene at each of various fixed scales and combine the results, as in scale space approaches [Mokhtarian et al., 2003] [Lindeberg, 1994]. Local scale approaches in image processing define the largest homogeneous region at each point, and treat these as fundamental units of the image. A similar global versus local dichotomy exists for describing shapes. To vary the level of detail depending on application, it is desirable to be able to detect dominant points on shape boundaries at different scales. In this chapter, we compare global, local, and locally adaptive scale approaches to shape analysis. For global scale, the Curvature Scale Space (CSS) [Mokhtarian et al., 2003] [Lindeberg, 1994] [Rattarangsi and Chin, 1992] and the adaptive corner detector based on CSS [Ray and Pandyan, 2003] methods are selected. CSS is a state-of-the-art shape descriptor, and is used in the MPEG-7 standard. The local scale method chosen is an angle-based approach [Rosenfeld and Johnston, 1973], which has been widely used in the literature. The locally adaptive scale approach is based on the notion of curvature-scale (c-scale), which is a new, locally adaptive scale concept, introduced in Chapter 3, that brings the idea of local morphometric scale (such as ball-, tensor-, and generalized scale) [Saha et al., 2000] [Madabhushi et al., 2006] developed for images to the realm of shape boundaries.

In this chapter, we present a thorough evaluation of these global and local scale methods of shape description. We use medical and non-medical data sets to perform qualitative and quantitative comparisons. We will demonstrate that locally adaptive scale has advantages over global scale in shape description, just as it has also been demonstrated in image filtering, segmentation, and registration. We first describe the image data sets in Section 4.1. The global and local scale methods used for comparison are briefly described in Section 4.2. In Section 4.3, we present a quantitative comparison of these methods. The qualitative results are shown in Section 4.4. Our conclusions are stated in Section 4.5.

4.1 Description of Image Data Sets

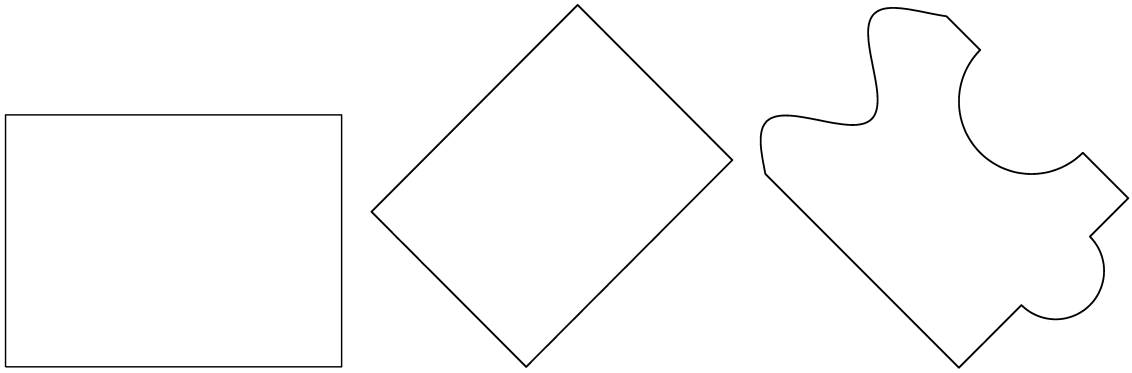
In this section, we present the data sets used to conduct comparisons at qualitative and quantitative levels. We have illustrated in the previous chapter the curve description process on a mathematical object (Fig.3.9). From now on, we will focus on the extraction of dominant points only. The data sets studied are divided into three main groups: mathematical and geometrical objects, natural objects, and medical objects. For the medical objects, we assume that the shapes are extracted after segmentation of appropriate images.

4.1.1 Mathematical and Geometrical Objects

We will use several mathematically constructed shapes and some geometrical digital shapes, where we can easily and accurately identify dominant points on the boundary. This will help us in comparing among the methods as to how good they are in detecting the correct points and in locating them precisely.

The mathematical shapes, presented in Fig. 4.1, are constructed from theoretical functions: a rectangle (Fig.4.1(a)), a rotated rectangle (Fig.4.1(b)), and a puzzle piece (Fig.4.1(c)). The rectangle and the rotated rectangle are formed by straight lines. The puzzle piece was introduced in Chapter 3 and it includes straight lines, circular arcs of different radii, one concave and one convex, and a sine wave. For these three shapes, we can mathematically

determine which points are dominant and find their exact position on the boundary to conduct a quantitative evaluation of the methods.



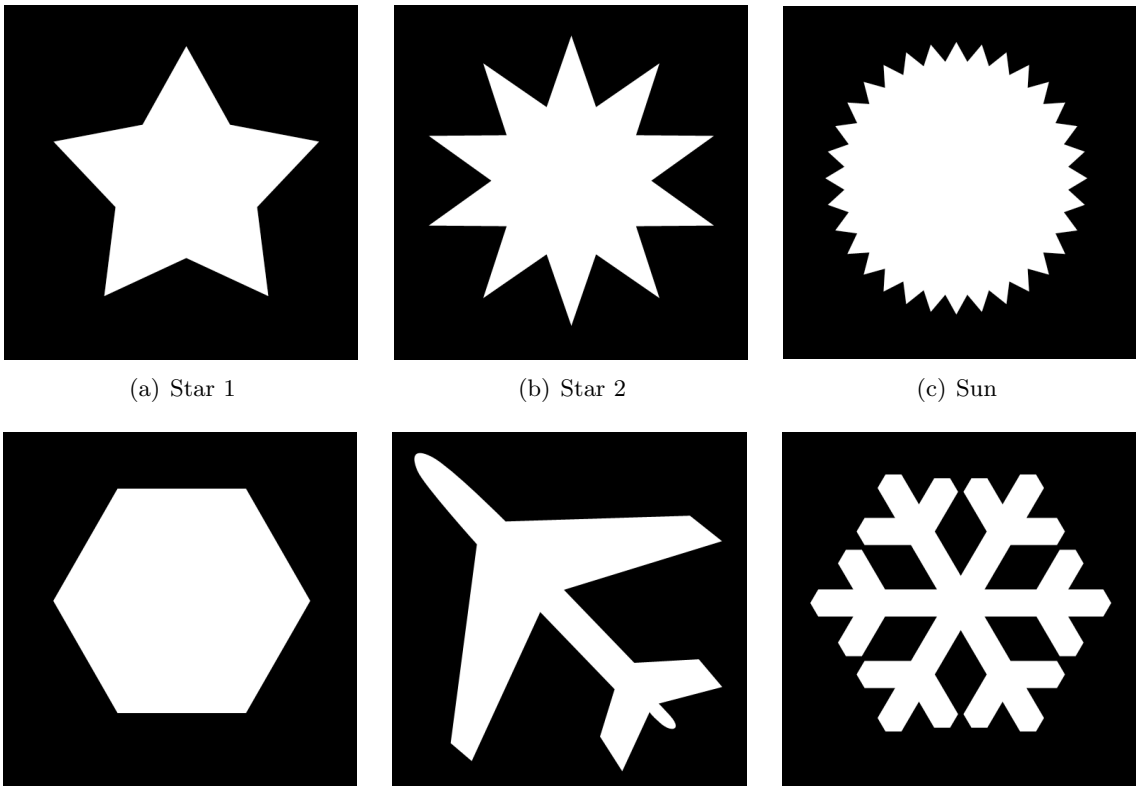
(a) Rectangle (b) Rotated rectangle (c) Puzzle piece

Figure 4.1: Mathematically constructed shapes.

We also used geometrical digital shapes (Fig. 4.2), which consist of a hexagon (Fig. 4.2(d)), two different star like shapes (Figs.4.2(a),4.2(b)), a sun (Fig.4.2(c)), an aeroplane (Fig.4.2(e)), and a snow flake pattern (Fig. 4.2(f)). All these shapes have different number of dominant points and varying characteristics. Dominant points can be easily located manually on these objects for the purpose of quantitative evaluation.

4.1.2 Natural Objects

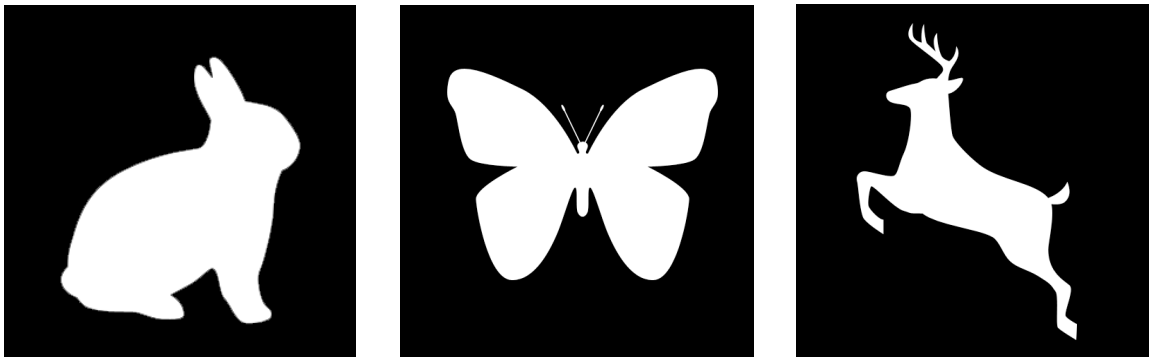
We selected non-medical objects that are natural and intuitive [Attneave, 1954] enough to visually identify dominant points and qualitatively assess the results obtained by different methods. We used a total of 6 objects with different characteristics and levels of detail, as shown in Fig. 4.3. The objects used are: a rabbit (Fig. 4.3(a)), a butterfly (Fig. 4.3(b)), a deer (Fig. 4.3(c)), a leaf (Fig. 4.3(d)), a fish (Fig. 4.3(e)), and a hand (Fig. 4.3(f)). We will perform a qualitative comparison of methods on the digital boundaries extracted from these binary objects.



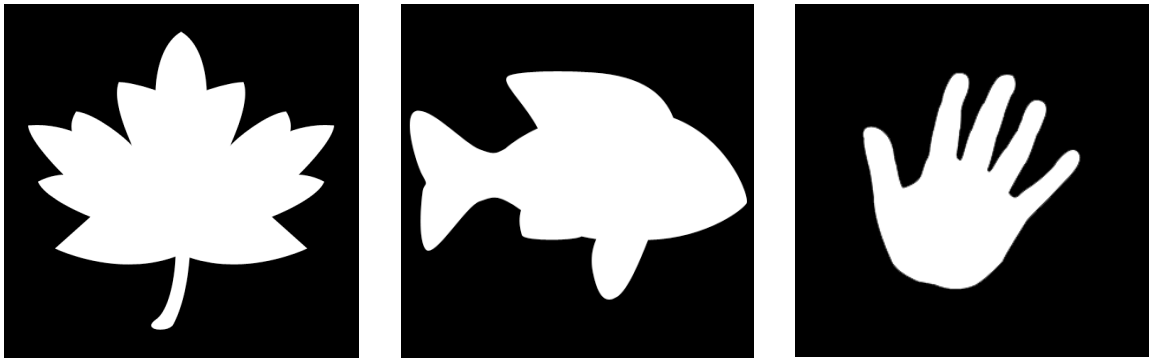
(a) Star 1 (b) Star 2 (c) Sun

(d) Hexagon (e) Aeroplane (f) Snow flake

Figure 4.2: Digital objects.



(a) Rabbit (b) Butterfly (c) Deer

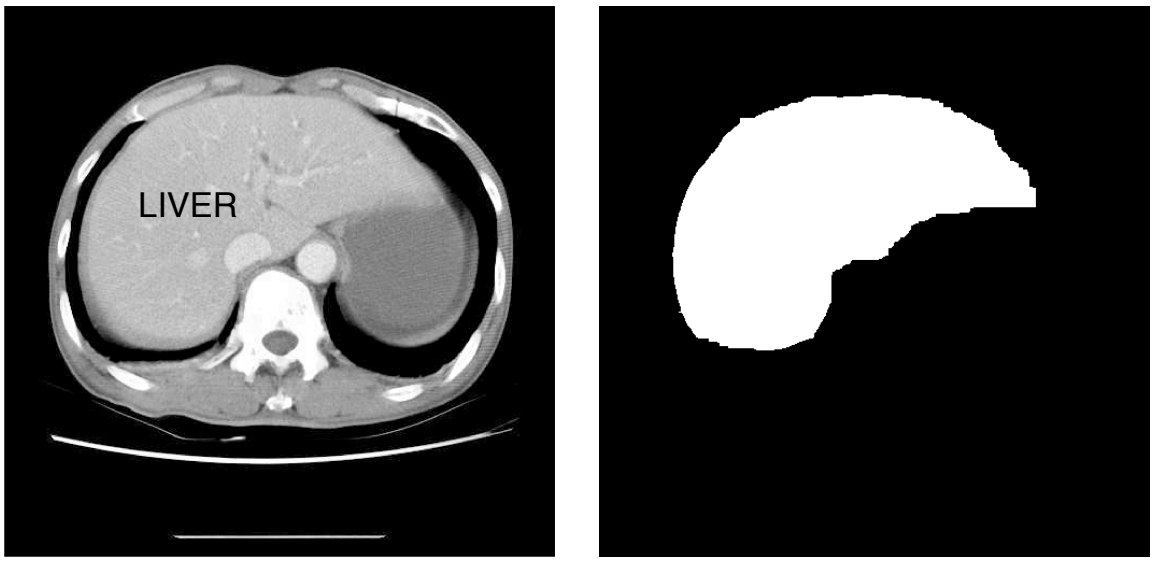


(d) Leaf (e) Fish (f) Hand

Figure 4.3: Natural objects.

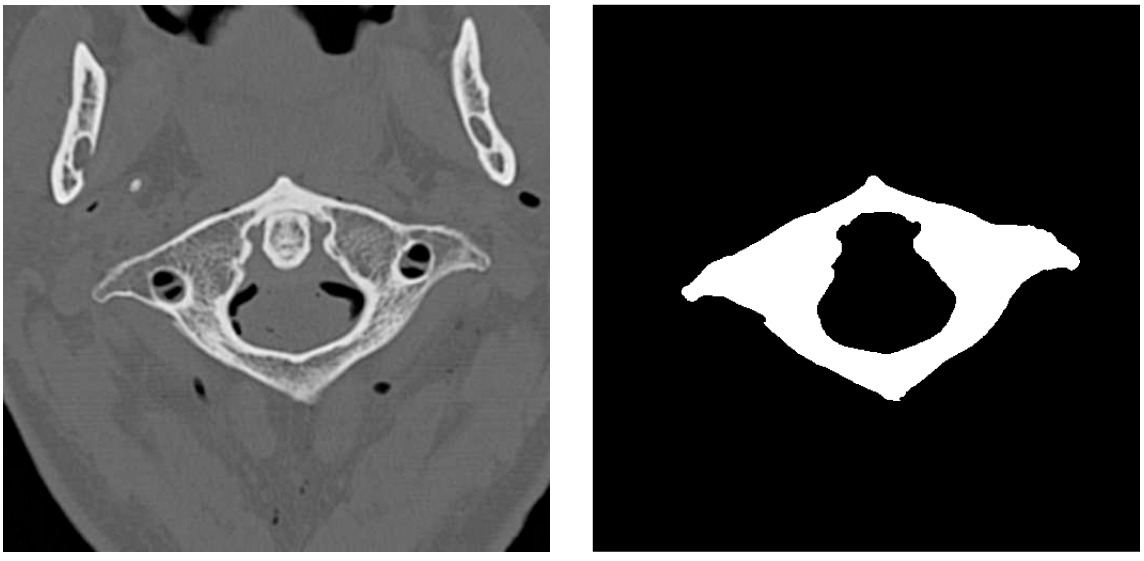
4.1.3 Medical Objects

We studied a total of six medical objects derived from both medical CT and MR images: the liver (Fig. 4.4), segmented from an abdominal CT image; a vertebra (Fig. 4.5), segmented from a spine CT image; talus and calcaneus bones, extracted from MR foot images (Fig. 4.6); brain cortex and brain ventricles (Fig. 4.7) segmented from brain MR images. The original images and the segmented objects are presented below. All the objects of interest were segmented by using an operator-steered Live Wire [Falcão et al., 2000] technique.



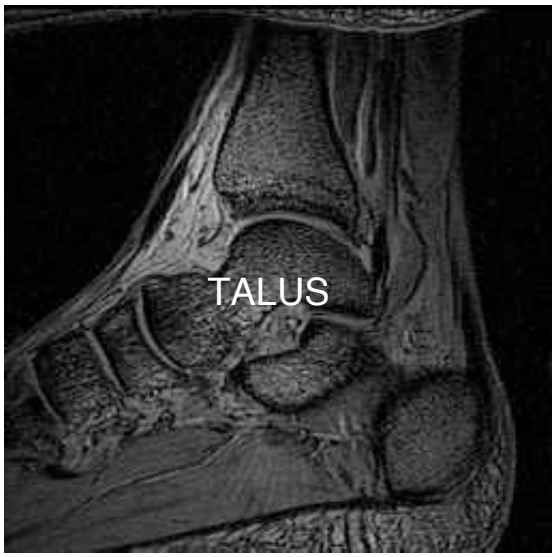
(a) Liver in a CT image (b) Segmented liver

Figure 4.4: Abdominal CT image.



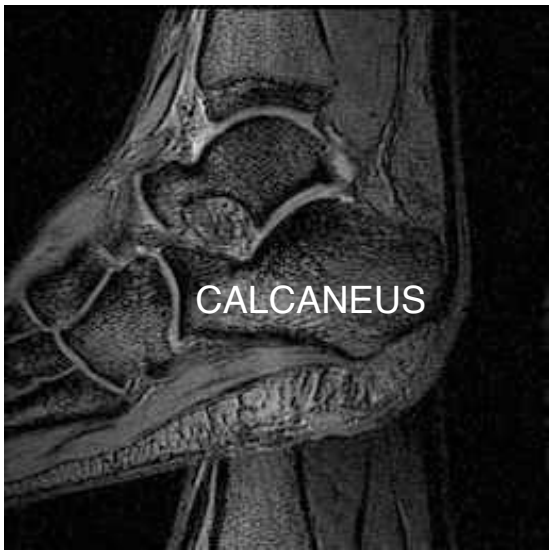
(a) Cervical vertebra in a CT image (b) Segmented cervical vertebra

Figure 4.5: Spine CT image.



(a) Talus in a foot MR image

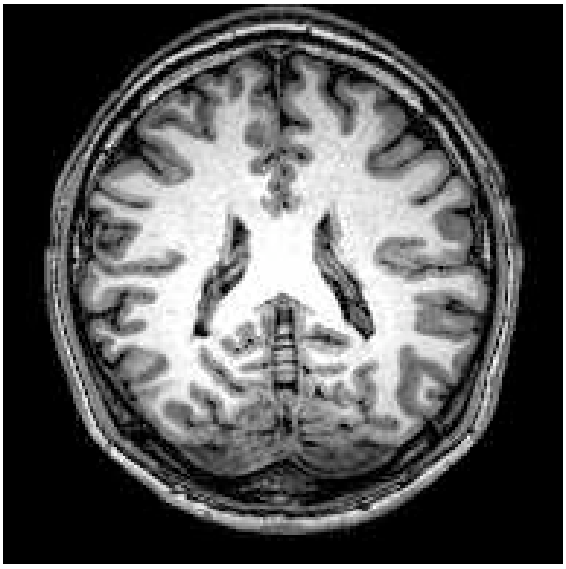
(b) Segmented talus



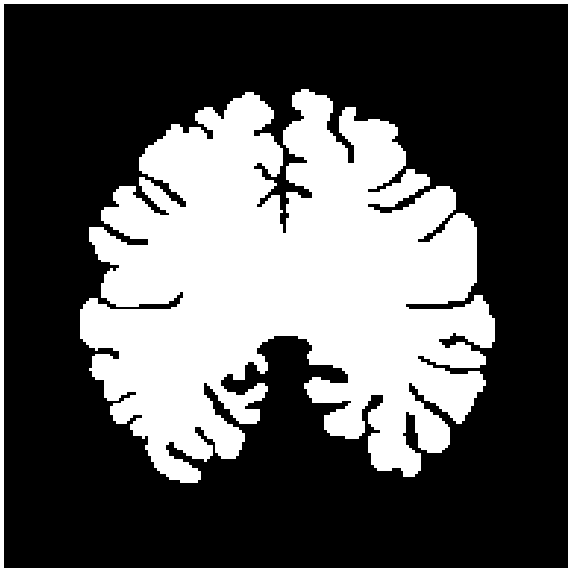
(c) Calcaneus in a foot MR image

(d) Segmented calcaneus

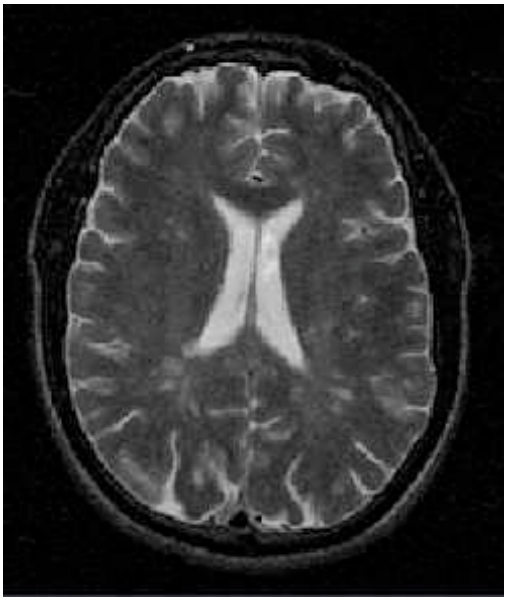
Figure 4.6: Foot MR image. (a-b) Talus. (c-d) Calcaneus.



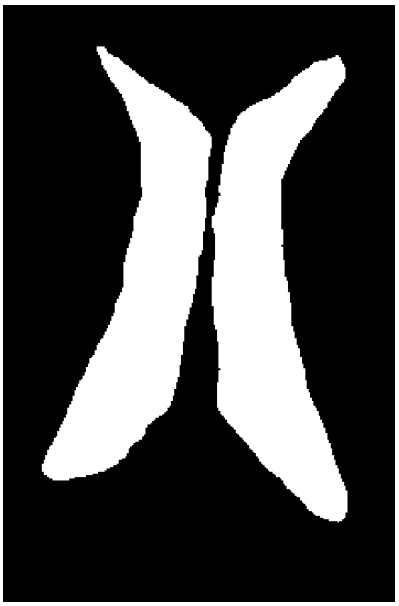
(a) Brain MR image depicting the cortex



(b) Segmented brain cortex



(c) Brain MR image depicting the ventricles



(d) Segmented brain ventricles

Figure 4.7: Brain MR image.

4.2 Comparison with Other Methods

In this section, we describe three methods, one using a local approach and two using global scale concepts, to compare with our locally adaptive scale approach presented in Chapter 3. The local scale approach is an angle-based shape descriptor [Rosenfeld and Johnston, 1973] [Rosenfeld and Weszka, 1975], commonly used to detect corners by the estimation of curvature in digital boundaries. The first global scale based approach utilized here is a state-of-the-art shape descriptor called CSS [Mokhtarian et al., 2003] [Rattarangsi and Chin, 1992] [Ray, B.K and Ray, K.S, 1995][Ray and Pandyan, 2003][Lindeberg, 1994] that has been used as a shape descriptor in the MPEG-7 standard [Mokhtarian et al., 2003]. Derived from this is the second global method, which is an adaptive corner detector (ACORD) based on CSS [Ray and Pandyan, 2003]. ACORD goes a step further in determining the scale that is most appropriate for each point on the boundary. This strategy was also selected for comparison purposes because its implementation is simpler than the CSS approach, avoiding the use of many rules that can create errors in the detection of dominant points. This analysis will allow us to evaluate and compare global and local scale approaches to shape description, but will also allow us to determine where our proposed approach stands in relation to what is considered as the state-of-the-art approaches.

4.2.1 Angle-Based Shape Descriptor

The angle-based approach [Rosenfeld and Johnston, 1973] [Rosenfeld and Weszka, 1975] is widely used in the literature [Neumann and Teisseron, 2002] [Ray, B.K. and Ray, K., 1992] [Davis, 1977] [Carmona et al., 2005] for extracting significant curvature maxima and minima directly on digital curves. The curvature at each bel b_i is estimated by considering the cosine of the angle formed by the vectors $\mathbf{A}_{ik} = \overrightarrow{b_i b_{i-k}}$ and $\mathbf{B}_{ik} = \overrightarrow{b_i b_{i+k}}$ (Fig. 4.8). The k-vectors at b_i are defined as $\mathbf{A}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$ and $\mathbf{B}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$. Consequently, the k-cosine $\mathbf{c}_{ik}$ at b_i is

$$\mathbf{c}_{ik} = \cos\Omega = \frac{\mathbf{A}_{ik} \cdot \mathbf{B}_{ik}}{\| \mathbf{A}_{ik} \| \| \mathbf{B}_{ik} \|}, \tag{4.1}$$

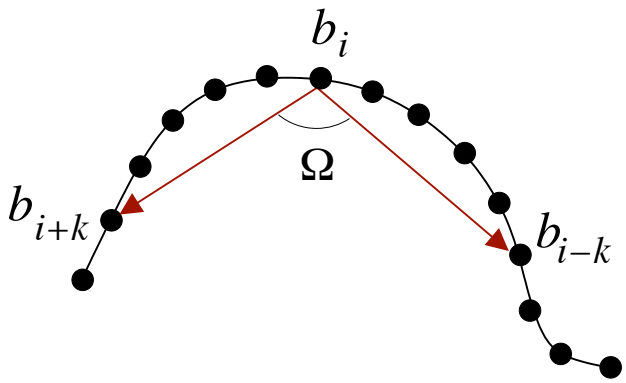


Figure 4.8: Angle-based approach for curvature estimation.

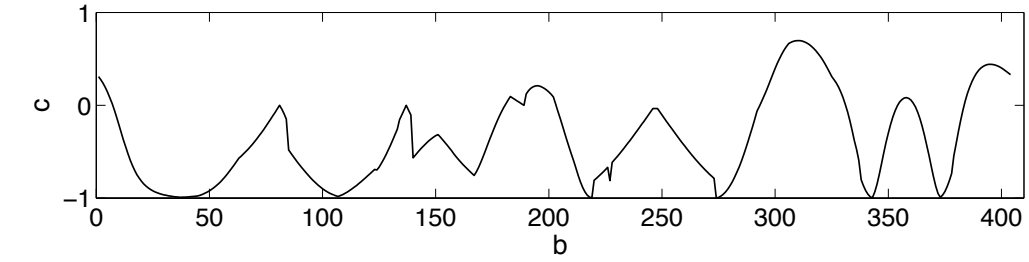
where $-1 \leq \mathbf{c}_{ik} \leq 1$. $\mathbf{c}_{ik}$ is close to 1 when the curve is turning rapidly, and close to -1 when the curve is relatively straight. The estimated curvature depends on the value of k. Therefore, an appropriate value of k at each point in the curve needs to be found. Let q be the total number of points forming the digital curve. An arbitrary value v, so that $\mu = \frac{q}{v}$, is set to compute $\mathbf{c}_{i1}, \mathbf{c}_{i2}, ..., \mathbf{c}_{i\mu}$ for each b_i. The optimal $\mathbf{c}_{i\eta}$ is selected such that

$$\mathbf{c}_{i\mu} < \mathbf{c}_{i,\mu-1} < \cdots < \mathbf{c}_{i\eta} \not< \mathbf{c}_{i,\eta-1}. \tag{4.2}$$

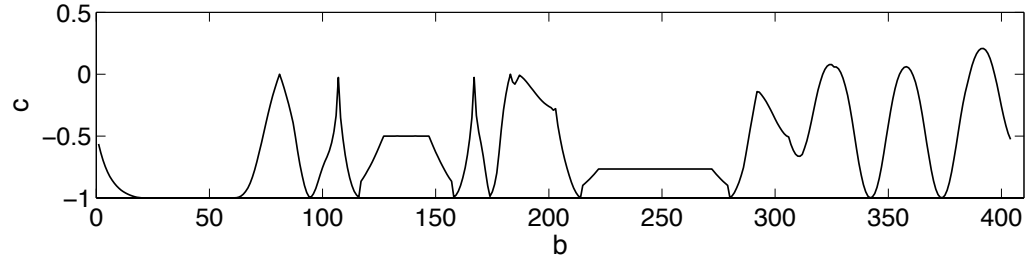
Each b_i has an associated $\mathbf{c}_{i\eta}$, denoted $\mathbf{c}_i$. The local curvature maxima are selected at b_i if $\mathbf{c}_i \geqslant \mathbf{c}_j$ for all j such that $\mid i - j \mid \leqslant \frac{\eta}{2}$. A similar operation is applied to detect curvature minima on the curve. To find dominant points, the only parameter we need to vary is v.

Influence of the parameter v

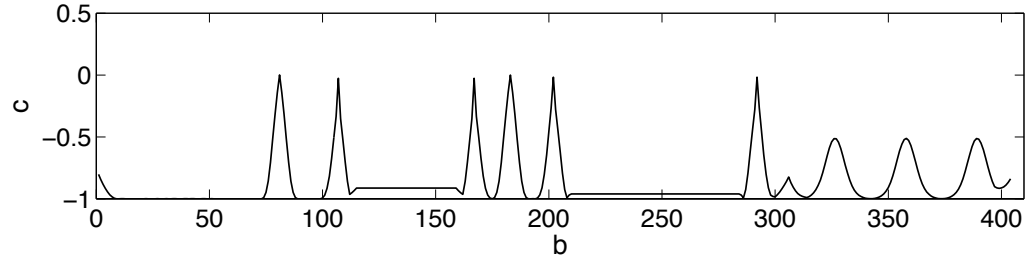
To assess how the choice of v influences the detection of dominant points, we use the mathematical shape in Fig. 3.9. In Figs. 4.9(a) - 4.9(c), we show the estimated curvature plots obtained for $v = 9$, $v = 20$, and $v = 50$, respectively, for the example of Fig. 3.9. The estimated curvature plots are highly dependent on the parameter v used, having a more detailed representation for a higher value of v (smaller μ) in the example shown. The peaks appearing in the curve correspond to corners in the boundary, and the valleys correspond to inflection points. If we select these minima and maxima of curvature according to the rule defined previously, we obtain Figs. 4.10(a)-4.10(c). We observe that the number of dominant points



(a) $v = 9$



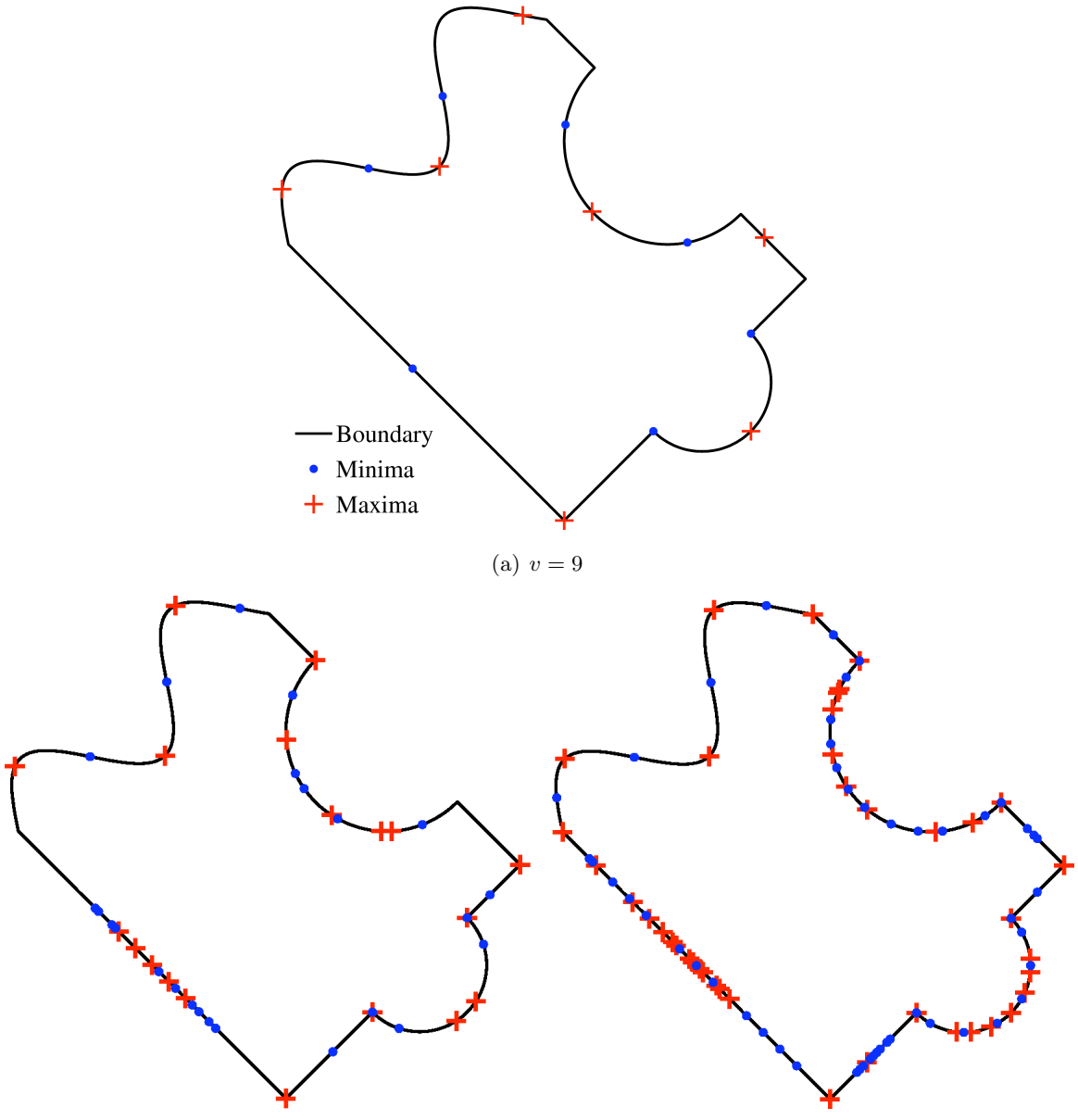
(b) $v = 20$



(c) $v = 50$

Figure 4.9: Angle-based curvature c for the shape in Fig.3.9 as a function of the parameter v.

detected increases as v increases. However, the number of spurious points becomes higher as well, especially in straight and circular regions. The number of real dominant points detected also increases for larger values of v. Therefore, this method is very sensitive to the parameter v, and a small variation of it can return very different results. A compromise is necessary to select a value for v that detects as many dominant points as possible with the minimal number of spurious corners and inflection points.



(a) $v = 9$

(b) $v = 20$

(c) $v = 50$

Figure 4.10: Detection of minima (blue dots) and maxima (red crosses) for the shape in Fig. 3.9.

4.2.2 Shape Description Via Curvature Scale Space

Global approaches, such as CSS [Rattarangsi and Chin, 1992] [Ray, B.K and Ray, K.S, 1995] [Mokhtarian et al., 2003] [Lindeberg, 1994], convolve the parametric boundary functions in both x and y with a varying scale Gaussian smoothing function. For each scale, the maxima of absolute curvature of the boundary are obtained and the scale space representation is established. The main idea of this type of methods is to detect dominant points at different scales and to use the information in all the scales as a whole, to decide whether a point is a corner or not.

Curvature Scale Space Representation

Let $\gamma(s) = (x(s), y(s))$ be the parametric representation of a planar curve, where s is the arc length parameter. The parameterized curvature $\kappa(s)$ can be defined, in terms of the derivative of the functions $x(s)$ and $y(s)$, as

$$\kappa(s) = \frac{\ddot{x}(s)\dot{y}(s) - \dot{x}(s)\ddot{y}(s)}{(\dot{x}(s)^2 + \dot{y}(s)^2)^{\frac{3}{2}}}, \tag{4.3}$$

where $\dot{x}(s)$ and $\ddot{x}(s)$ denote the first and second derivatives of $x(s)$, respectively. Similarly, $\dot{y}(s)$ and $\ddot{y}(s)$ represent the first and second derivatives of $y(s)$.

To obtain the curvature at different scales, the concept of *curve evolution* is used, which can be achieved by convolving both $x(s)$ and $y(s)$ by a smoothing Gaussian function $g(s, \sigma)$, defined as

$$g(s, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}}, \tag{4.4}$$

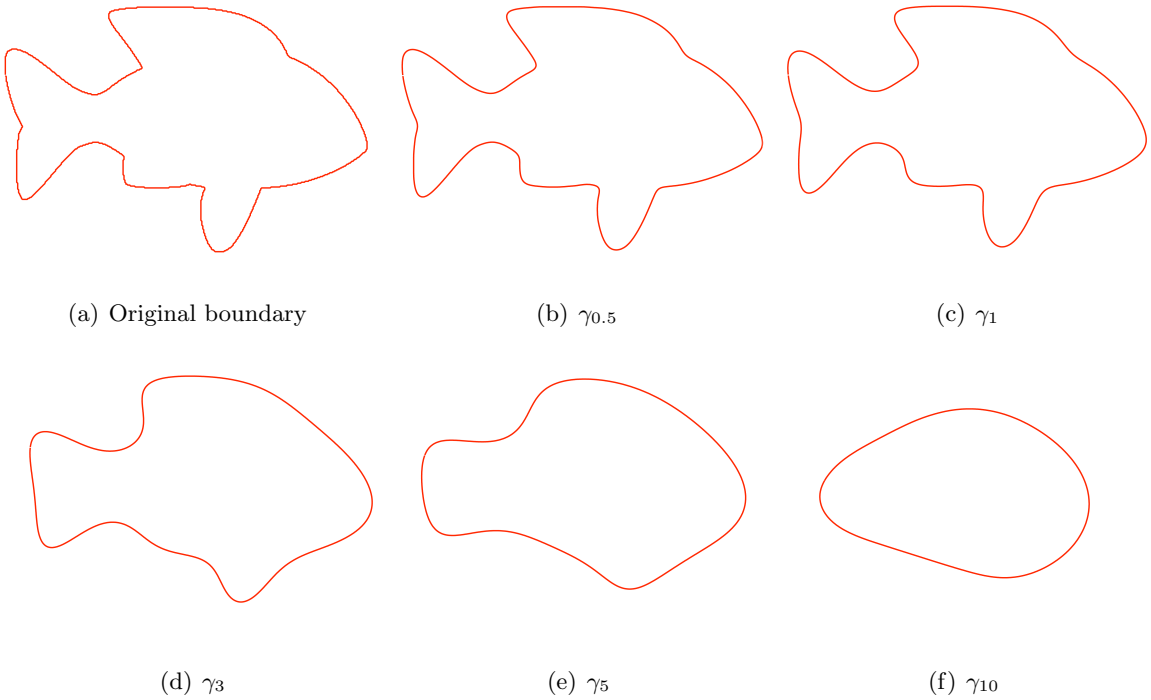
where σ is the standard deviation of the Gaussian and the scale parameter. The convolution of x and y with the Gaussian kernel of width σ is given by

$$X(s, \sigma) = x(s) * g(s, \sigma) = \int_{u=s-S/2}^{u=s+S/2} x(u) g(s - u, \sigma) du \tag{4.5}$$

and

$$Y(s,\sigma) = y(s) * g(s,\sigma) = \int_{u=s-S/2}^{u=s+S/2} y(u)g(s-u,\sigma)du, \tag{4.6}$$

where S is the total length of the curve. $X(s,\sigma)$ and $Y(s,\sigma)$ are the new components of the evolved curve, denoted γ_σ. An example of curve evolution for the boundary in Fig. 4.11(a) is shown in Figs. 4.11(b)-4.11(f).



(a) Original boundary (b) $\gamma_{0.5}$ (c) γ_1

(d) γ_3 (e) γ_5 (f) γ_{10}

Figure 4.11: Curve evolution illustrated on the fish boundary of (a) for (b) $\sigma = 0.5$, (c) $\sigma = 1$, (d) $\sigma = 3$, (e) $\sigma = 5$, and (f) $\sigma = 10$.

The convolution with the Gaussian kernel smoothes the curve, also reducing the effects of noise. It is assumed that the curve is closed and that σ is small with respect to the length of the curve to avoid aliasing effects.

According to the properties of convolution, the derivatives $\dot{X}(s,\sigma)$ and $\ddot{X}(s,\sigma)$ are given by

$$\dot{X}(s,\sigma) = x(s) * \dot{g}(s,\sigma) \tag{4.7}$$

and

$$\ddot{X}(s,\sigma) = x(s) * \ddot{g}(s,\sigma), \tag{4.8}$$

where $\dot{g}(s,\sigma)$ and $\ddot{g}(s,\sigma)$ denote the first and second derivatives of $g(s,\sigma)$, respectively. Similar formulas exist for $\dot{Y}(s,\sigma)$ and $\ddot{Y}(s,\sigma)$. Therefore, the curvature of the evolved curve can be expressed as

$$\kappa(s,\sigma) = \frac{\dot{X}(s,\sigma)\ddot{Y}(s,\sigma) - \dot{Y}(s,\sigma)\ddot{X}(s,\sigma)}{(\dot{X}(s,\sigma)^2 + \dot{Y}(s,\sigma)^2)^{\frac{3}{2}}}. \tag{4.9}$$

Curvature Scale Space Map

For each scale σ, the goal is to determine the dominant points on the curve γ_σ. The corners or high curvature points correspond to the maxima of absolute curvature, $\mid \kappa(s,\sigma) \mid$, on the boundary, which are the positive maxima and negative minima of curvature. Therefore, we need to find the points in γ_σ that satisfy for $\kappa(s,\sigma) > 0$,

$$\frac{\partial \kappa(X(s,\sigma), Y(s,\sigma))}{\partial s} = 0 \qquad \text{with} \qquad \frac{\partial^2 \kappa(X(s,\sigma), Y(s,\sigma))}{\partial s^2} < 0, \tag{4.10}$$

and for $\kappa(s,\sigma) < 0$,

$$\frac{\partial \kappa(X(s,\sigma), Y(s,\sigma))}{\partial s} = 0 \qquad \text{with} \qquad \frac{\partial^2 \kappa(X(s,\sigma), Y(s,\sigma))}{\partial s^2} > 0. \tag{4.11}$$

The detected absolute maxima of curvature at each scale σ are plotted as a scale space map. The x-axis of the map corresponds to the parameter s, and the y-axis corresponds to the number of iterations of σ. The line patterns of the scale space map indicate the locations of the maxima through varying scales. The sign of $\kappa(s,\sigma)$, associated with each line pattern, indicates the concavity of the curve. Only certain line patterns can appear in the scale space map depending on the type of corners existing in the shape considered. An example of scale

space map is shown in Fig. 4.12 for the shape in Fig. 3.9. To control the number of iterations of σ to consider, we need to define σ_{min}, σ_{max}, and $\triangle\sigma$ to build the scale space map. $\triangle\sigma$ is the discrepancy between two consecutive σ values and it needs to be set small enough to create a scale space map with continuous line patterns.

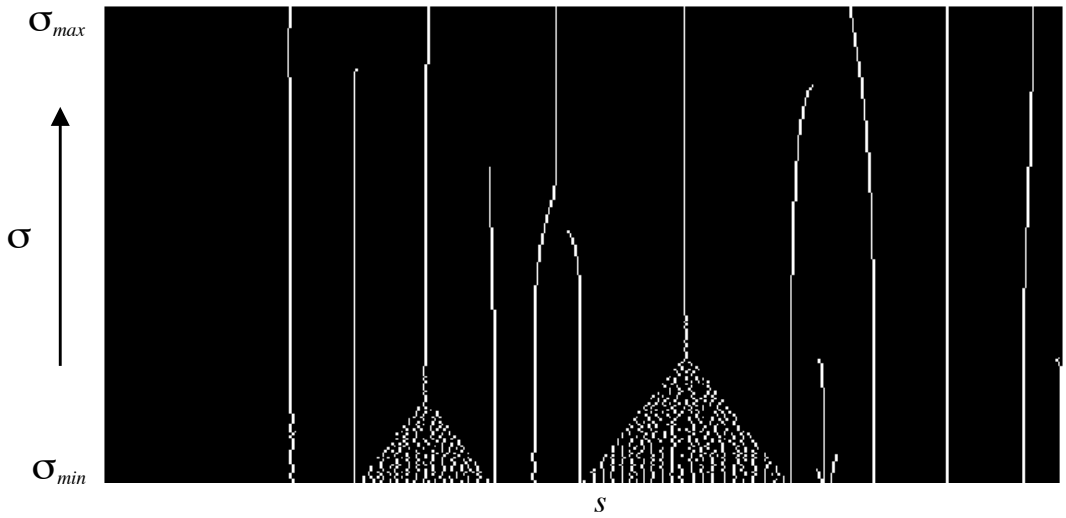


Figure 4.12: Scale space map for the shape in Fig. 3.9. $\sigma_{min} = 0.01$, $\sigma_{max} = 2$, and $\triangle\sigma = 0.01$. The triangular shapes of noise pattern correspond to the circular segments in Fig. 3.9.

There are basically three types of models used to analyse the scape space patterns: Γ model, STAIR model, and END model [Rattarangsi and Chin, 1992]. Each model has a characteristic pattern that facilitates its identification. Each line pattern must persist, repel, terminate, attract, or merge with a neighbouring line following certain rules that allow the characterization of a corner.

Detection of Dominant Points

The scale space map is transformed into a set of trees [Rattarangsi and Chin, 1992] to simplify the representation and to detect relevant dominant points. To build the trees, it is necessary to know the *persistency*, *sign*, and *movement* for each maximum of curvature de-

tected on the boundary [Rattarangsi and Chin, 1992] . Persistence and movement can be established by tracking the patterns on the scale space map. If a line pattern survives more than 50% of the iteration scale, then a vertical line is drawn from the finest scale to the largest iteration scale for which it survives. If it does not survive 50% of the iteration scale, the sign and movement of the pattern are checked to see if the pattern can be linked to the nearest persistent line with a horizontal line. The sign of the extreme curvature generating the dot pattern is also necessary and it indicates the concavity of the curve. If a persistent line and a non-surviving dot pattern have the same sign and attract each other, it is indicative of an END model and can be represented that way in the tree. If the non-surviving dot pattern and the persistent line have opposite signs and repel each other, then it is an indication of a STAIR model. The trees corresponding to the scale space map in Fig. 4.12 are shown in Fig. 4.13.

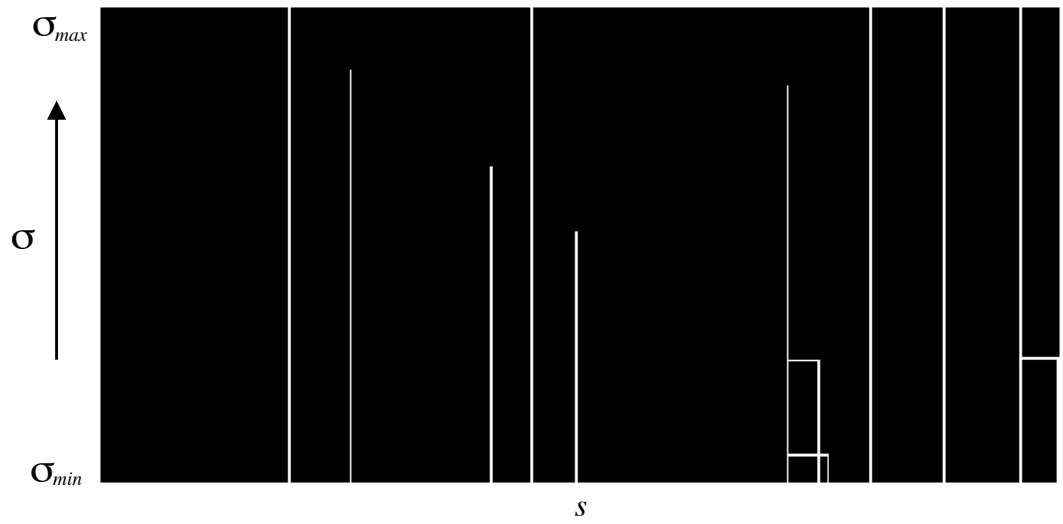


Figure 4.13: Trees extracted from the scale space map in Fig. 4.12 for the shape in Fig. 3.9. $\sigma_{min} = 0.01$, $\sigma_{max} = 2$, and $\triangle\sigma = 0.01$.

Once the trees are built, corners are identified at multiple scales in a coarse-to-fine manner. The selection of corners is based on a set of rules that attempts to assure a stability criterion. The stability is determined by measuring the length of corresponding roots, branches, and leaves of the trees, excluding the absolute maxima caused by digital effects and noise. The

corners detected for the shape in Fig. 3.9 and corresponding to the scale space map in Fig. 4.12 and the trees in Fig. 4.13 are shown in Fig. 4.14.

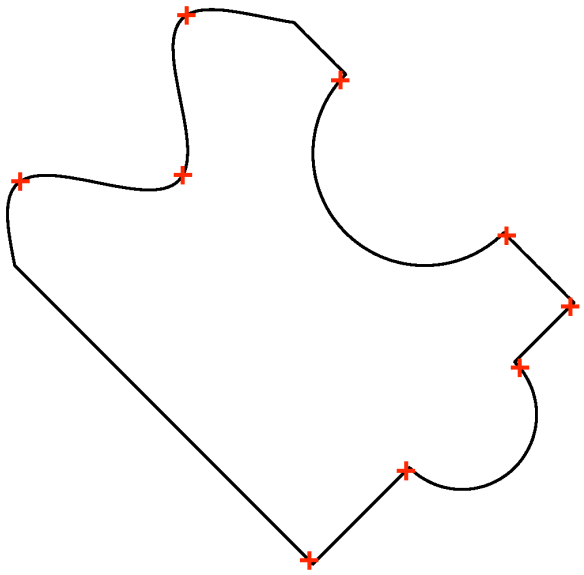


Figure 4.14: Dominant points detected from the trees in Fig. 4.13 for the shape in Fig. 3.9. $\sigma_{min} = 0.01$, $\sigma_{max} = 2$, and $\triangle\sigma = 0.01$.

To vary the number of corners detected, we can tune the parameters σ_{min} and σ_{max}. Generally, to increase the number of corners detected, we can decrease σ_{max}. Similarly, to decrease the number of corners detected, we increase σ_{max}. Normally, σ_{min} is set as low as possible to eliminate the noise in the boundary during the extraction of corners without affecting the accuracy in the detection of corners, as the location of the corners is always identified at the bottom of the tree. In most of the examples shown in this chapter, we set σ_{min} to 0.1. $\triangle\sigma$ is set to 0.01 in all the examples used in order to have continuous line patterns in the scale space map to achieve a correct tracking of the lines.

4.2.3 Adaptive Corner Detector Based on Curvature Scale Space

One of the main problems of the CSS strategy is the selection of the appropriate scale to detect all the corners existing in a shape. Furthermore, the construction of the scale space map and conversion into trees are ad-hoc processes subjected to many rules. To

overcome these drawbacks, the adaptive corner detector (ACORD) strategy was developed [Ray and Pandyan, 2003]. This approach consists of a global scale strategy based on CSS that adaptively selects the scale to use at each bel b_i of a discrete curve, while avoiding the creation of a scale space map or construction of trees. Then, by using the calculated scale at each bel, the curvature is estimated at each b_i and the corners of the shape are extracted by identifying the absolute local maxima on the curvature plot.

For a bel b_i in $\mathcal{B}$, with coordinates (x_i, y_i), we can define the first order discrete derivatives as

$$\triangle x_i = \frac{x_{i+1} - x_{i-1}}{\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}}, \tag{4.12}$$

$$\triangle y_i = \frac{y_{i+1} - y_{i-1}}{\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}}, \tag{4.13}$$

and the second order discrete derivatives as

$$\triangle^2 x_i = \frac{\frac{x_{i+1}-x_i}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}} - \frac{x_i-x_{i-1}}{\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2}}}{\frac{1}{2}\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}}, \tag{4.14}$$

$$\triangle^2 y_i = \frac{\frac{y_{i+1}-y_i}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}} - \frac{y_i-y_{i-1}}{\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2}}}{\frac{1}{2}\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}}. \tag{4.15}$$

With these formulas, we can represent the curvature κ_i at each point b_i as

$$\kappa_i = \triangle x_i \, \triangle^2 y_i - \triangle y_i \, \triangle^2 x_i. \tag{4.16}$$

In the original CSS method [Mokhtarian et al., 2003] [Rattarangsi and Chin, 1992], the whole curve was smoothed at all levels of detail, whereas in this approach, the level of smoothing is adaptively determined for each bel b_i by measuring the roughness of the curve at a neighbourhood of b_i. The roughness of the curve is represented by the variance in curvature in the neighbourhood of each b_i. The higher the variance measured in a neighbourhood of

b_i, the higher is the roughness associated with it.

The curvature at any level of smoothing is calculated by convolving the curve by the coefficients of a discrete Gaussian kernel of window size W. The coefficients for $W = 3$ are defined as

$$g_{-1} = 0.2236,\ g_0 = 0.5477,\ \text{and}\ g_1 = 0.2236. \tag{4.17}$$

To obtain the coefficients for higher sizes of W, we convolve the coefficients of a window of size $W - 2$, W being an odd number, with the coefficients of $W = 3$.

The convolution of x_i and y_i with the Gaussian kernel of width W is given by

$$X_i = \sum_{j=-\frac{W-1}{2}}^{j=\frac{W-1}{2}} g_j x_{i+j}, \tag{4.18}$$

and

$$Y_i = \sum_{j=-\frac{W-1}{2}}^{j=\frac{W-1}{2}} g_j y_{i+j}, \tag{4.19}$$

where X_i and Y_i are the smoothed components of the curve at b_i. By replacing x_i and y_i by X_i and Y_i in (4.12-4.16), we can calculate the curvature $\kappa_{i,W}$ at b_i for a particular size W of the Gaussian kernel. The variance of curvature $(\sigma_{i,W})^2$ at each bel for a particular level of smoothing W is defined as

$$(\sigma_{i,W})^2 = \sum_{-\frac{W-1}{2}}^{\frac{W-1}{2}} \frac{(\kappa_{i,W})^2}{W} - \sum_{-\frac{W-1}{2}}^{\frac{W-1}{2}} (\frac{\kappa_{i,W}}{W})^2. \tag{4.20}$$

For a particular bel b_i, the necessary smoothing in the neighbourhood of b_i is attained at a size W if $(\sigma_{i,W})^2 \leq (\sigma_{i,W-2})^2$ is achieved for the first time. Then, $\kappa_{i,W}$ is the associated curvature value at that bel b_i. If $(\sigma_{i,W})^2 > (\sigma_{i,W-2})^2$, the window size is increased by 2 and the smoothing continues until the stopping criterion is met. To avoid aliasing effects, W should always be smaller than the length of the curve. The curvature obtained for the shape in Fig. 3.9 is represented in Fig. 4.15.

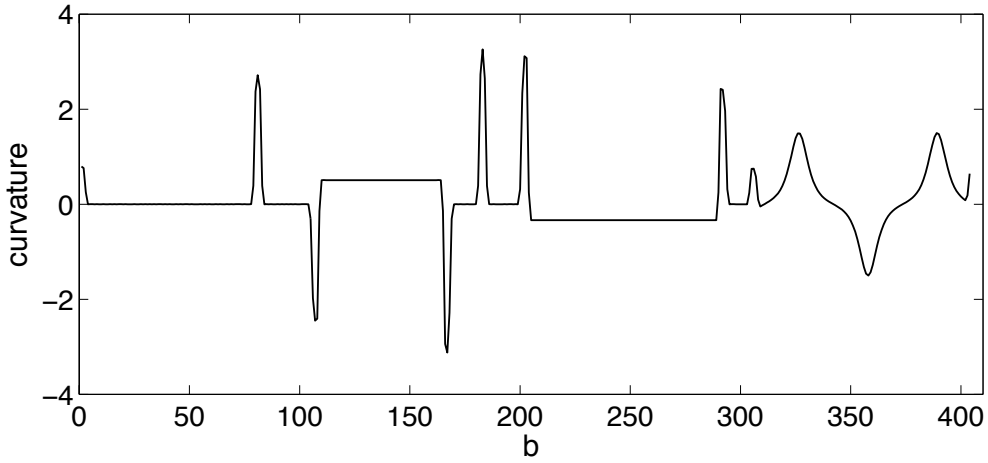


Figure 4.15: Curvature obtained with the ACORD method for the shape in Fig. 3.9.

The corners on the shape are selected by detecting the absolute local maxima of the curvature above a threshold T such as

$$|\kappa_{i,W}| \geq T. \tag{4.21}$$

The threshold T should be above the level of noise in the curvature plot to detect as many corners as possible and not spurious corners. The detected absolute local maxima for the shape in Fig. 3.9 with $T = 0.1$ are shown in Fig. 4.16.

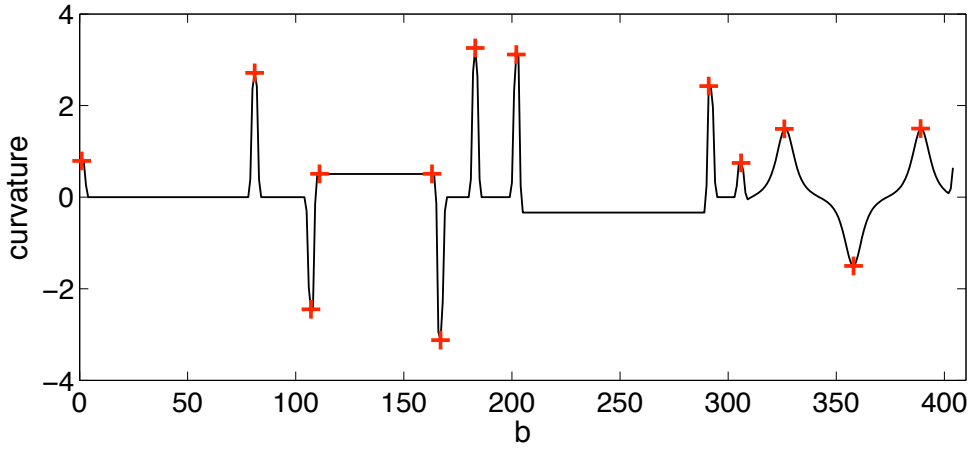


Figure 4.16: Detection of the absolute maxima of curvature with the ACORD method on the curvature plot for the shape in Fig. 3.9.

The last step of the method consists of merging all close corners to avoid representing the same corner more than once. From a cluster of close corners, we select the one having the highest curvature of all in a small neighbourhood. In Fig. 4.16, there are two close corners at each side of the first circular arc (the first constant region in Fig. 4.16 different from zero) that need to be merged.

To extract corners from the curvature plot, we have to carefully select the threshold T. A value of $T = 0.1$ seems to work well for most of the cases studied. We have merged all close corners that were at a distance smaller than 5 bels to avoid identifying the same corner more than once. The detected corners for the shape in Fig. 3.9 are shown in Fig. 4.17 after merging close corners.

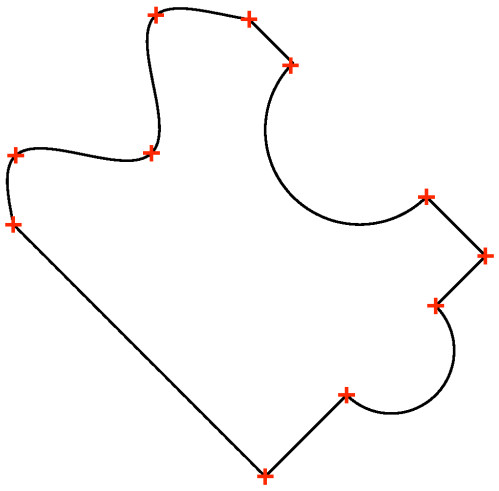


Figure 4.17: Corners detected with the ACORD method for the shape in Fig. 3.9.

It should be pointed out that this method is no longer multi-scale, as CSS was, since the optimal value of scale has been selected at each bel and all the scales are combined to obtain the curvature plot that best suits a particular curve, depending on its local curve roughness.

4.3 Quantitative Comparison

Quantitative evaluation is done in terms of three factors: the number of *false positives* (FPs), the number of *false negatives* (FNs), and *accuracy measure*. The FP factor indicates the

number of falsely identified dominant points on the boundary. The FN factor corresponds to the number of true dominant points that were missed on the boundary. Accuracy is measured in terms of the root mean square (RMS) location error, defined as

$$RMS = \sqrt{\frac{d_1^2 + d_2^2 + \cdots + d_n^2}{n}}, \tag{4.22}$$

where $d_1, d_2, \ldots, d_n$ are the distances between the true and detected dominant points for a certain method, n being the maximum number of dominant points detected with that approach. The RMS values are expressed as a percent of the length of the shape:

$$\%RMS = \frac{RMS \times 100}{ls}, \tag{4.23}$$

where ls denotes the length of the shape, measured as the maximum distance between any two points in the shape.

For quantitative analysis, we used a total of 9 objects, the three mathematically constructed, shown in Fig. 4.1 and the six digital geometrical objects, shown in Fig. 4.2, where dominant points can be unequivocally identified. Furthermore, the c-scale approach and the angle-based method can directly identify high curvature *and* inflection points on the contours, whereas the CSS and the ACORD methods detect only high curvature points. If we wish to detect the inflection points by using CSS, another algorithm should be implemented only for that particular task, whereas in the c-scale approach and the angle-based approach, both are obtained simultaneously. In this section, high curvature points are represented by a red cross and inflection points by a blue dot. In the following, we will quantitatively assess how these four methods perform for different objects with varying complexity.

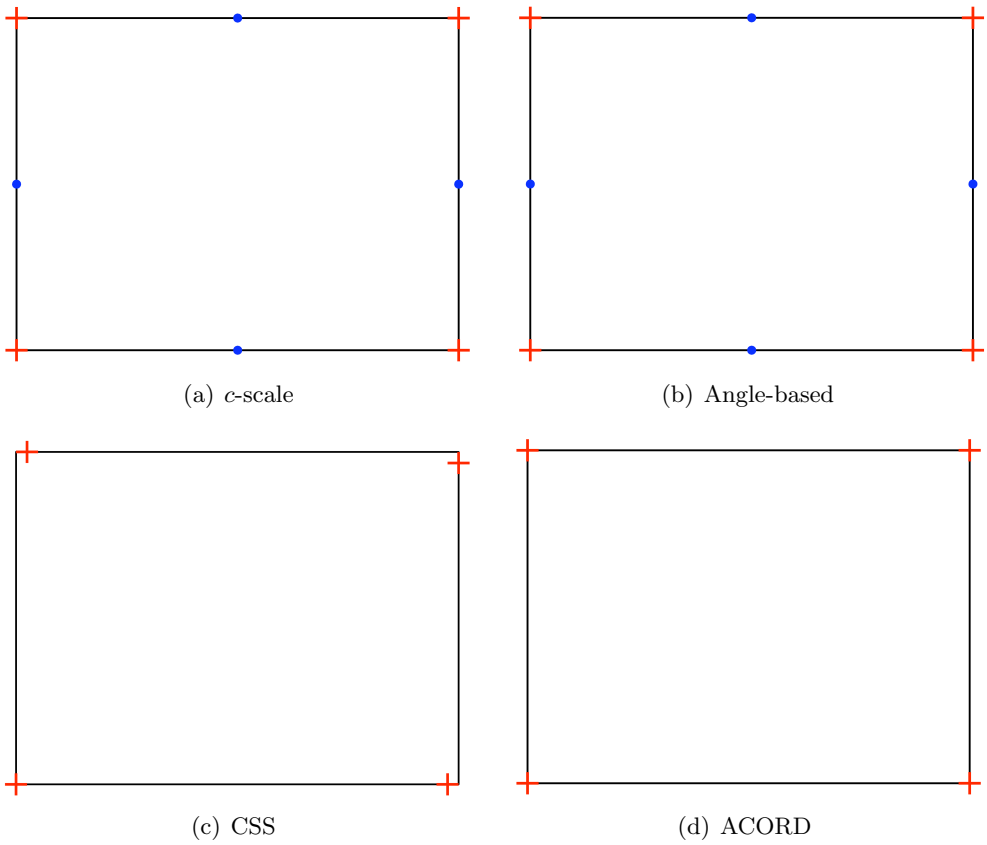
4.3.1 Quantitative Comparison of High Curvature Points

The quantitative evaluation of the four methods performed on the set of 9 objects is summarized in Table 4.1 in terms of FPs, FNs, the total number of points detected by each method, the real number of corners existing in the shape considered, and the % RMS error.

Table 4.1: Quantitative comparison of the methods for the detection of high curvature points by using nine known shapes.

Shape		Method	FPs	FNs	Points Detected	Total Points	RMS error (%)
Mathematical	Rectangle	c-scale	0	0	4	4	0
		Angle-based	0	0	4	4	0
		CSS	0	0	4	4	1.9999
		ACORD	0	0	4	4	0
	Rotated rectangle	c-scale	0	0	4	4	0
		Angle-based	0	0	4	4	0
		CSS	0	0	4	4	3.1623
		ACORD	0	0	4	4	0
	Puzzle piece	c-scale	0	0	11	11	0.4339
		Angle-based	2	6	7	11	3.8663
		CSS	0	0	11	11	1.3212
		ACORD	0	0	11	11	0.2311
Digital	Hexagon	c-scale	0	0	6	6	0.2239
		Angle-based	0	0	6	6	0
		CSS	4	0	10	6	0.5251
		ACORD	0	0	6	6	0.2742
	Star 1	c-scale	0	0	10	10	0.2804
		Angle-based	0	0	10	10	1.6910
		CSS	0	0	10	10	0.4631
		ACORD	0	3	7	10	0.3196
	Star 2	c-scale	0	0	20	20	0.2056
		Angle-based	0	0	20	20	0.1062
		CSS	0	0	20	20	0.4111
		ACORD	0	0	20	20	0.1839
	Sun	c-scale	0	0	64	64	0.2209
		Angle-based	2	32	34	64	4.6269
		CSS	0	0	64	64	0.4866
		ACORD	10	0	74	64	0.2087
	Aeroplane	c-scale	0	0	18	18	0.2337
		Angle-based	0	9	7	18	1.7534
		CSS	0	4	14	18	0.2952
		ACORD	2	3	17	18	0.2603
	Snow flake	c-scale	0	0	84	84	0.1926
		Angle-based	0	36	48	84	9.0026
		CSS	0	5	79	84	0.4416
		ACORD	0	0	84	84	0.2301

The dominant points detected by the four methods for the rectangle shape are displayed in Figs.4.18(a)-4.18(d). For this shape, according to Table 4.1, the c-scale, the angle-based,



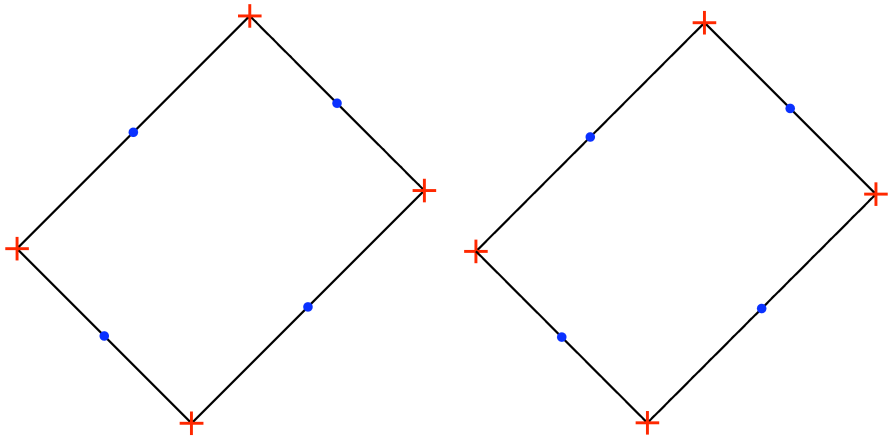
(a) c-scale (b) Angle-based

(c) CSS (d) ACORD

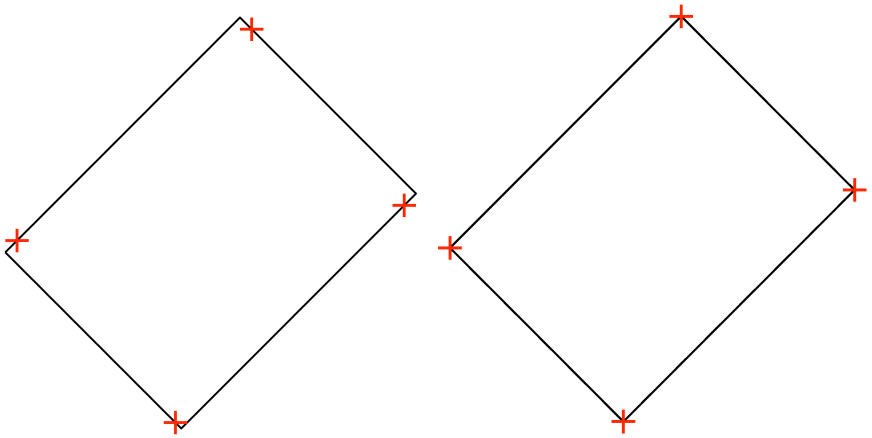
Figure 4.18: The dominant points detected for the rectangle shape (Fig. 4.1(a)) by (a) the c-scale method with $t = 0.02$, no smoothing of $A(b)$, and $se_v = se_p = 5$, (b) the angle-based method with $v = 5$, (c) the CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$, and (d) the ACORD method with $T = 0.1$.

and the ACORD methods perform similarly and find all four dominant points without any error, FPs, or FNs. However, the CSS method had an RMS error of around 2%, which can also be noticed in the accuracy of the localization of the points in Fig. 4.18(c), where the corners detected fall out of the real corners for three of them.

Similar to the rectangle shape, the corners detected on the rotated rectangle by c-scale (Fig. 4.19(a)), angle-based (Fig. 4.19(b)), and ACORD (Fig. 4.19(d)) methods perform perfectly whereas the CSS method presents an RMS error above 3%. The error in localization



(a) c-scale (b) Angle-based



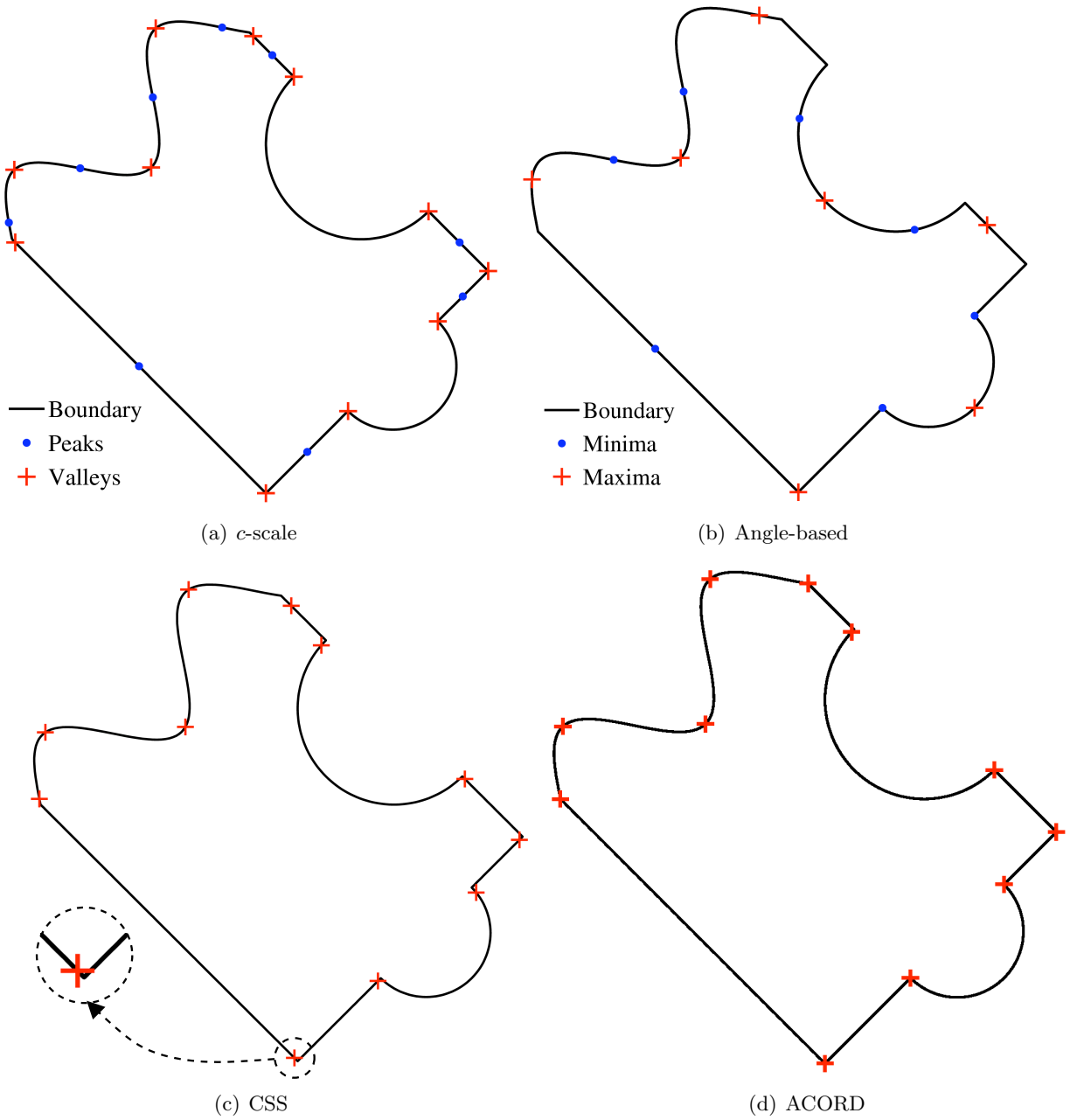
(c) CSS (d) ACORD

Figure 4.19: The dominant points detected for the rotated rectangle shape by (a) the c-scale method with $t = 0.02$, no smoothing of $A(b)$, and $se_v = se_p = 5$, (b) the angle-based method with $v = 5$, (c) the CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$, and (d) the ACORD method with $T = 0.1$.

for the CSS method can also be seen in Fig. 4.19(c), where the located corners are displaced from the real corners.

The results for the puzzle piece shape, displayed in Figs. 4.20(a)-4.20(d), show that, for the best parameter setting, the angle-based method is unable to detect all corners and falsely identifies some corners, as indicated by the FN and FP factors in Table 4.1. Furthermore, some unreasonable results appear, such as the detection of two minima in the corners of one of the circular arcs, and the detection of points in the circular arcs where the curvature is constant and no dominant points exist. Moreover, the inaccuracies in the locations of the corners, as displayed in Fig. 4.20(b) and also shown by the RMS errors in Table 4.1, indicate that the angle-based method has the highest localization error followed by the CSS approach. The inaccuracies for the CSS method can also be observed in the close-up view in Fig.4.20(c). The c-scale and ACORD approaches perform similarly in terms of localization error, the ACORD method being more accurate in this case. Overall for this shape, the detection of corners by using c-scale and ACORD methods is more robust and accurate than with the CSS and angle-based approaches.

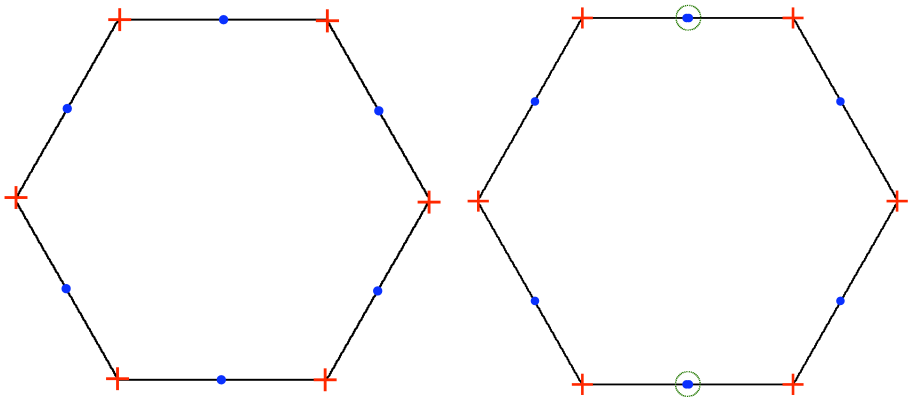
Now we come to the digital shapes. For the hexagon shape, the results are shown in Figs. 4.21(a)-4.21(d) and in Table 4.1. The angle-based method performs better than any method tested for this shape, achieving an RMS error of 0% for the detection of corners, identifying only the true corners of the shape. However, this method is not very robust since a small variation of v can completely alter the results. Instead of $v = 9$ in Fig. 4.21(b), if we use $v = 10$, already two falsely identified corners appear close to the true corners on each side of the shape. c-scale has a very small error and is closely followed by the ACORD method in terms of localization accuracy. CSS performs worse than any of the methods tested with a higher RMS error and 4 falsely identified corners.



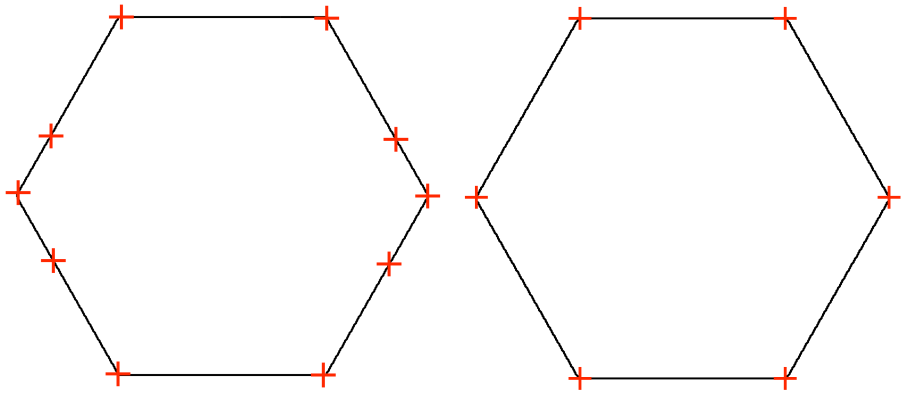
(a) c-scale (b) Angle-based

(c) CSS (d) ACORD

Figure 4.20: The dominant points detected for the shape in Fig.3.9 by (a) the c-scale method with $t = 0.02$, no smoothing of $A(b)$, and $se_v = se_p = 5$, (b) the angle-based method with $v = 9$, (c) the CSS method (a close-up view is also shown) with $\sigma_{min} = 0.01$ and $\sigma_{max} = 1$, and (d) the ACORD method with $T = 0.1$.



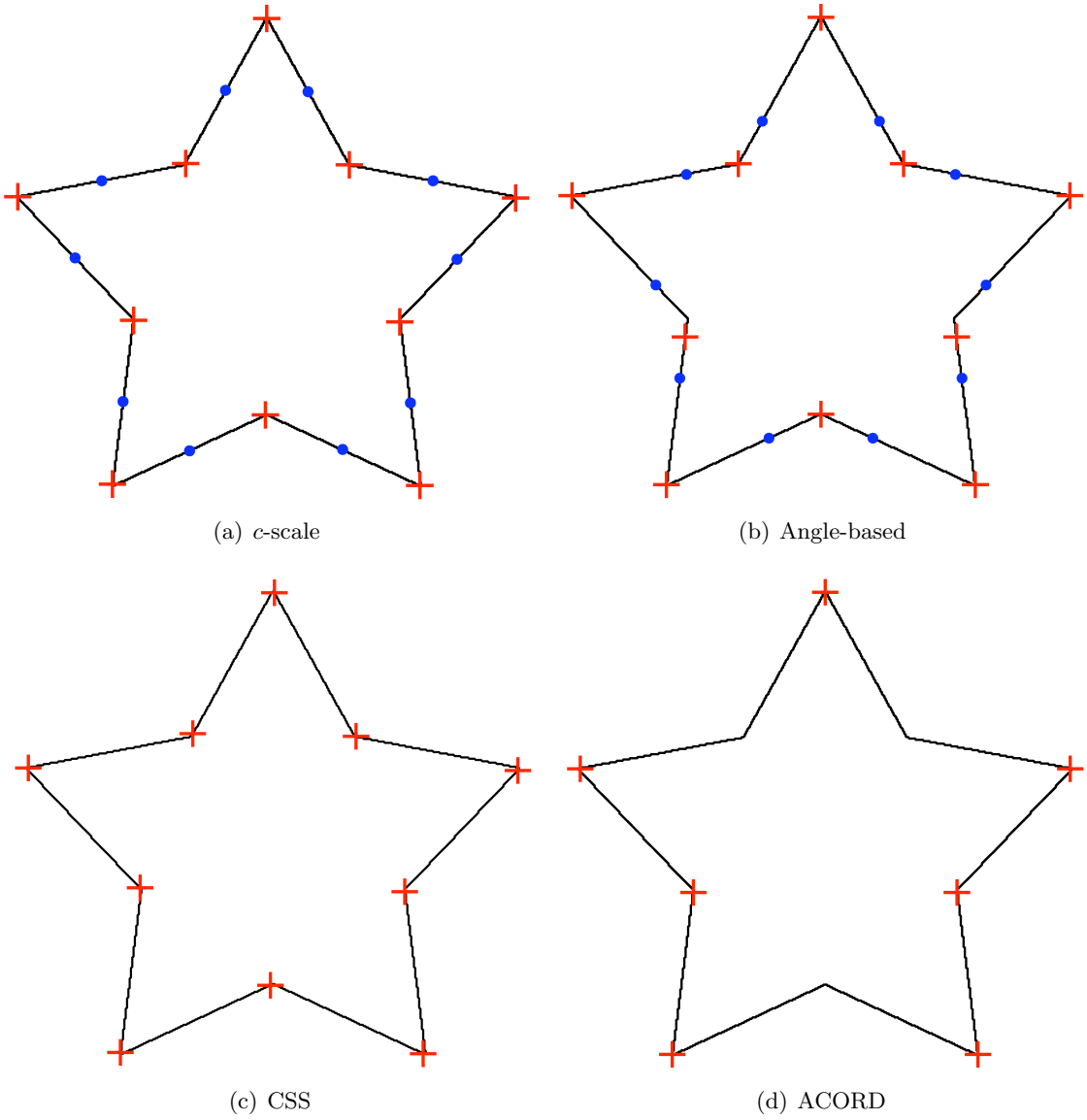
(a) c-scale (b) Angle-based



(c) CSS (d) ACORD

Figure 4.21: The dominant points detected on the hexagon shape. (a) c-scale method with $t = 1$, $w = 1$, $m = 1$, $se_v = se_p = 5$. (b) Angle-based method with $v = 9$. The green circle indicates that there are two points detected very close at that location. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (d) ACORD $T = 1$.

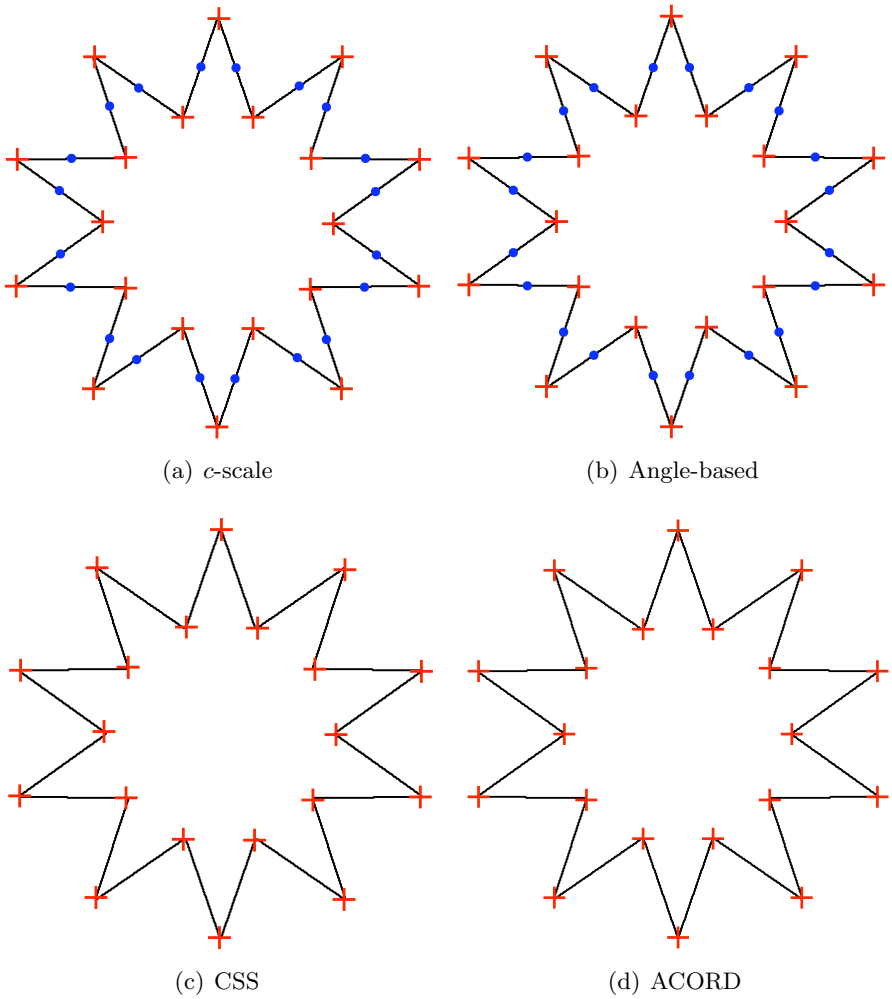
For the star shape, the results presented in Figs. 4.22(a)-4.22(d) and in Table 4.1 indicate that the c-scale, angle-based, and CSS methods find only the true corners existing in the shape whereas the ACORD approach misses 3 corners. For localization accuracy, the best approach is the c-scale method. It is closely followed by the ACORD and the CSS approaches. The worst method in terms of RMS error is the angle-based.



(a) c-scale (b) Angle-based

(c) CSS (d) ACORD

Figure 4.22: The dominant points detected on the star 1 shape. (a) c-scale method with $t = 1$, $w = 1$, $m = 1$, $se_v = se_p = 5$. (b) Angle-based method with $v = 9$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (d) ACORD $T = 1$.



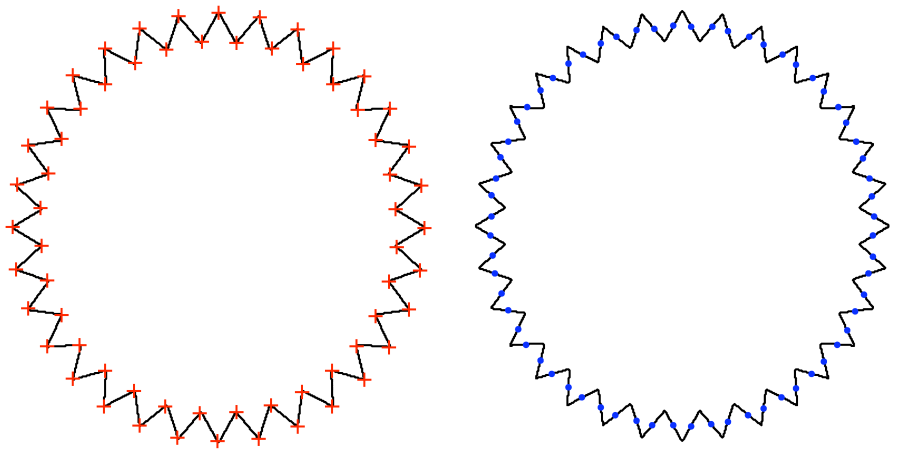
(a) c-scale (b) Angle-based

(c) CSS (d) ACORD

Figure 4.23: The dominant points detected on the star 2 shape. (a) c-scale method with $t = 1$, $w = 1$, $m = 1$, $se_v = se_p = 5$. (b) Angle-based method with $v = 20$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 6$. (d) ACORD $T = 0.1$.

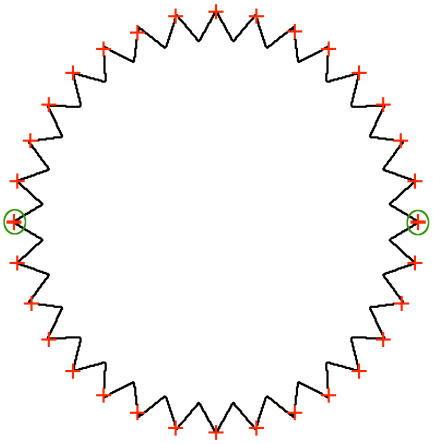
The results for the star shape can be seen in Figs. 4.23(a)-4.23(d) and in Table 4.1. All methods find all and only the true corners existing in this shape. The localization error (0.1%) is best for the angle-based approach. c-scale and ACORD have similar errors of around 0.2%, whereas CSS performs worst with an RMS error of 0.4%.

The results for the sun shape are presented in Figs. 4.24(a)-4.24(f).

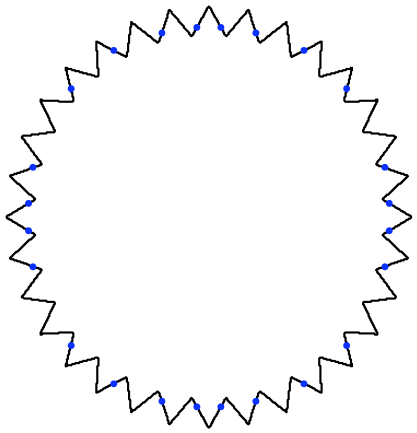


(a) c-scale valleys

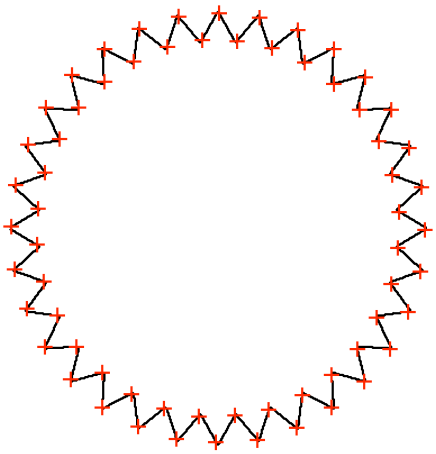
(b) c-scale peaks



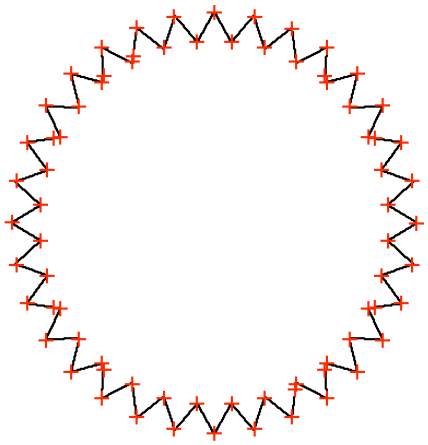
(c) Angle-based maxima



(d) Angle-based minima



(e) CSS



(f) ACORD

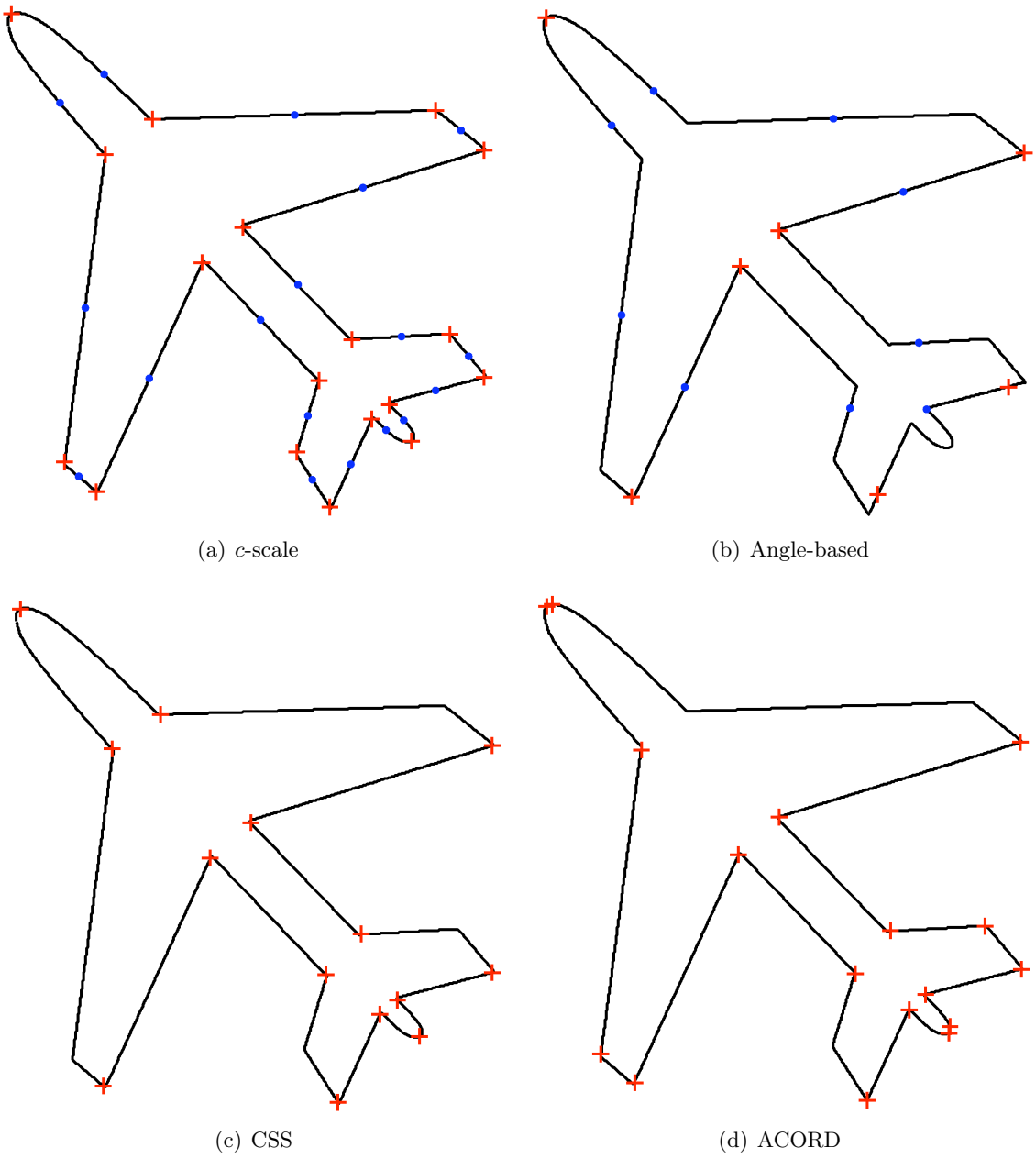
Figure 4.24: The dominant points detected on the sun shape. (a) c-scale method with $t = 1$, $w = 1$, $m = 1$, $se_v = se_p = 5$. (b) Angle-based method with $v = 20$. The green circle indicates that there are two points detected at that location. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (d) ACORD $T = 0.1$.

Overall, c-scale performs better than any other method for this shape by capturing all and only the true corners while having one of the smallest RMS errors among all compared methods. The ACORD approach has a slightly smaller RMS error than c-scale, however, it falsely identifies 10 corners that do not exist in the real shape, as can be noticed in Fig. 4.24(f). CSS also finds all and only the true corners, but has twice as much RMS error as c-scale. The angle-based approach is the worst for this shape as it falsely identifies 2 corners and misses other 32, while having a high RMS error of 4.63%. The falsely identified corners can be seen in Fig. 4.24(c) encircled in green.

The detected corners for the aeroplane shape are shown in Figs. 4.25(a)-4.25(d). All methods studied, except c-scale, have falsely identified corners or missed corners in the shape. The best method in terms of RMS error (0.23%) is also the c-scale method. The CSS and ACORD approaches have slightly higher RMS errors and the worst localization error happens for the angle-based approach, with a value of 1.75%. Overall, c-scale performs the best on the aeroplane shape and detects all true corners accurately.

In the snow flake shape, the c-scale and the ACORD approaches identify all and only the true corners as shown in Figs.4.26(a) and 4.26(f), as well as in Table 4.1, whereas the CSS and the angle-based approaches miss some of the corners (Figs. 4.26(c) and4.26(e)), as indicated by the FN factors. Additionally, c-scale presents a lower RMS error in the locations themselves of the correctly identified points, whereas the angle-based approach reaches an RMS value of 9% for the corners detected. Overall, c-scale performs better than any other method for the snow flake shape.

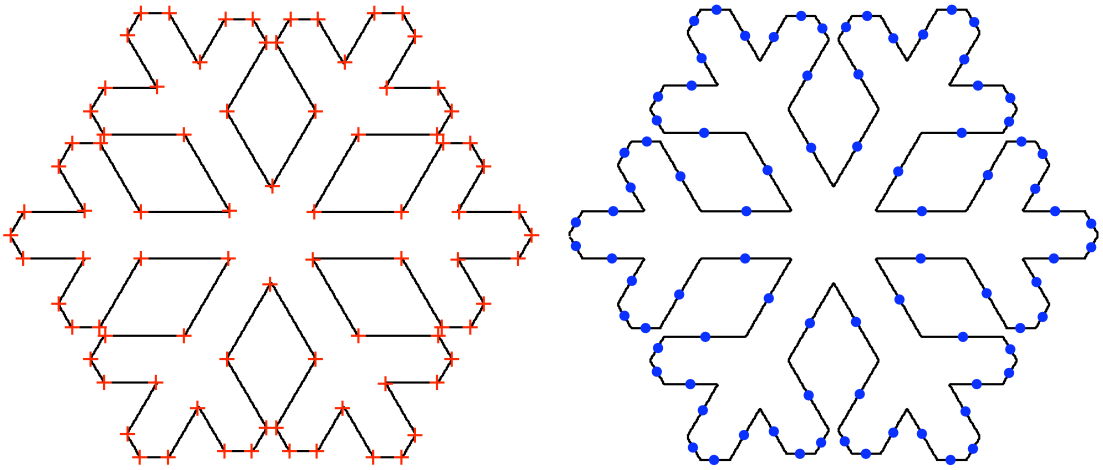
From the results in Table 4.1 and Figs. 4.18-4.26, we can conclude that c-scale is more robust and accurate than any other method studied as it did not detect FPs or FNs in any of the shapes studied while presenting small RMS errors for all shapes. All other methods encountered false detections or missed certain corners for some of the shapes. Furthermore, c-scale has, for almost all the digital shapes studied, an RMS error around 0.2% independent



(a) c-scale (b) Angle-based

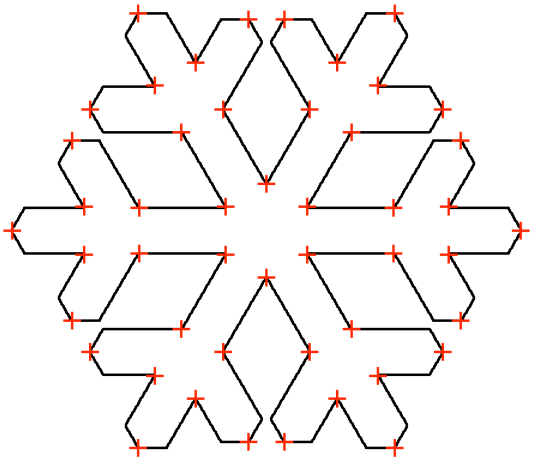
(c) CSS (d) ACORD

Figure 4.25: The dominant points detected on the plane shape. (a) c-scale method with $t = 2$, $w = 2$, $m = 2$, $se_v = se_p = 5$. (b) Angle-based method with $v = 13$. The green circle indicates that there are two points detected at that location. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 6$. (d) ACORD $T = 0.1$.

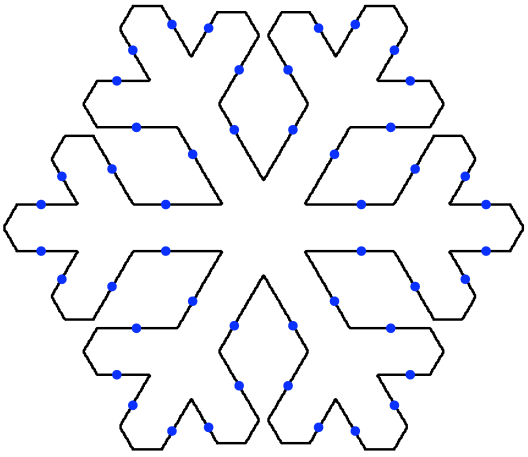


(a) c-scale valleys

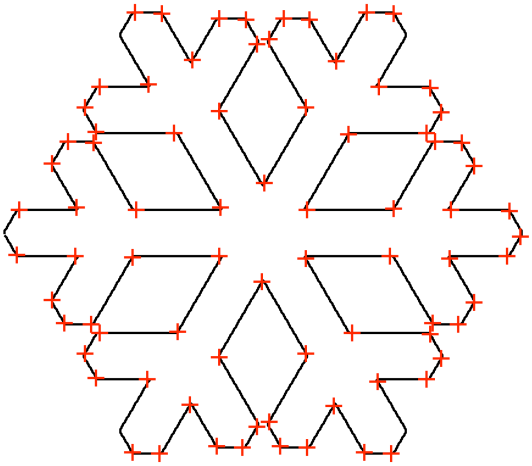
(b) c-scale peaks



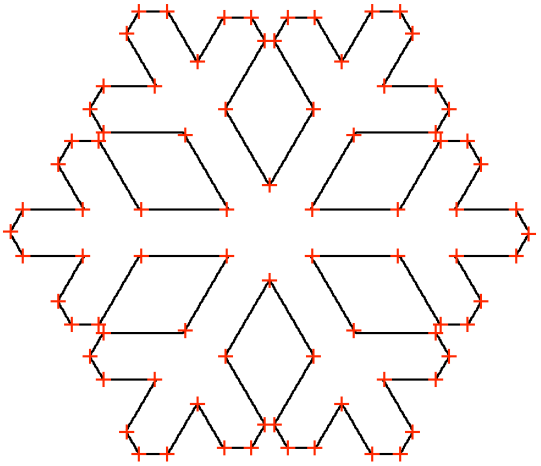
(c) Angle-based maxima



(d) Angle-based minima



(e) CSS



(f) ACORD

Figure 4.26: The dominant points detected on the snow shape. (a-b) c-scale method with $t = 1.1$, $w = 1$, $m = 1$, $se_v = se_p = 5$. (c-d) Angle-based method with $v = 66$. (e) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 2$. (f) ACORD $T = 0.1$.

of the shape considered. This indicates that the detection of corners is consistent and does not depend on the type of the shape. For the angle-based approach, the RMS error varied significantly from shape to shape, which makes the detection of corners highly dependent on the type of shape considered with this approach. The CSS method has RMS values around $0.4\% - 0.5\%$, which is also consistent among shapes, but presents a higher variance in the results than c-scale. The ACORD approach also exhibits a low RMS error around $0.2\% - 0.3\%$, which also indicates that the method is quite independent of the type of shape used. However, it demonstrates a good number of FPs and FNs.

4.3.2 Quantitative Comparison of Inflection Points

The results of the quantitative comparison for the detection of inflection points on the 9 shapes are presented in Table 4.2. Among the four methods studied, only the c-scale and angle-based approaches can find inflection points. The displays in Figs. 4.18-4.26 already demonstrated the inflection points detected by the two methods.

For the rectangle and the rotated rectangle, both methods find the inflection points perfectly with 0% RMS error and without FPs or FNs. For all the other shapes, c-scale performs significantly better in terms of RMS error with values lower than 0.23% for the digital shapes. The angle-based approach has in almost all the shapes studied, an RMS error greater than 1%, reaching even 17% for the snow flake shape (Fig. 4.26(d)). The smallest RMS error presented for the angle-based method is 0.76% for the star 2 shape (Fig. 4.23(b)). We can visually appreciate the error in most of the shapes, such as in star 1 (Fig. 4.22(b)), where the inflection points do not fall in the middle of the straight segments as they do for the c-scale approach (Fig. 4.22(a)). Furthermore, c-scale does not encounter any FPs or FNs in any of the shapes studied, whereas the angle-based approach falsely identifies inflection points in the puzzle piece (Fig. 4.20(b)), the hexagon (Fig. 4.21(b), indicated by a green circle), and the aeroplane (Fig. 4.25(b)). It also misses some inflection points in the puzzle piece (Fig. 4.20(b)), the aeroplane (Fig. 4.25(b)), the sun (Fig. 4.24(d)), and the snow flake (Fig. 4.26(d)).

Table 4.2: Quantitative evaluation of the methods for the detection of inflection points by using the 9 known shapes.

Shape		Method	FPs	FNs	Points Detected	Total Points	RMS error(%)
Mathematical	Rectangle	c-scale	0	0	4	4	0
		Angle-based	0	0	4	4	0
	Rotated rectangle	c-scale	0	0	4	4	0
		Angle-based	0	0	4	4	0
	Puzzle piece	c-scale	0	0	9	9	1.1773
		Angle-based	4	6	7	9	2.2173
Digital	Hexagon	c-scale	0	0	6	6	0.1583
		Angle-based	2	0	8	6	1.8121
	Star 1	c-scale	0	0	10	10	0.1891
		Angle-based	0	0	10	10	6.3623
	Star 2	c-scale	0	0	20	20	0.2123
		Angle-based	0	0	20	20	0.7618
	Aeroplane	c-scale	0	0	18	18	0.2275
		Angle-based	1	10	9	18	2.3752
	Sun	c-scale	0	0	64	64	0.2030
		Angle-based	0	40	24	64	1.5627
	Snow flake	c-scale	0	0	84	84	0.2090
		Angle-based	0	36	48	84	17.0723

Exactly as for corner detection, c-scale for digital shapes has almost a constant RMS error of about 0.2% independent of shapes. Therefore, c-scale is consistent not only in the detection of corners, but also in the detection of inflection points, with the same accuracy rate and among shapes, independently of their complexity, which makes c-scale a robust and accurate method to use for detecting dominant points of any kind.

4.4 Qualitative Comparison

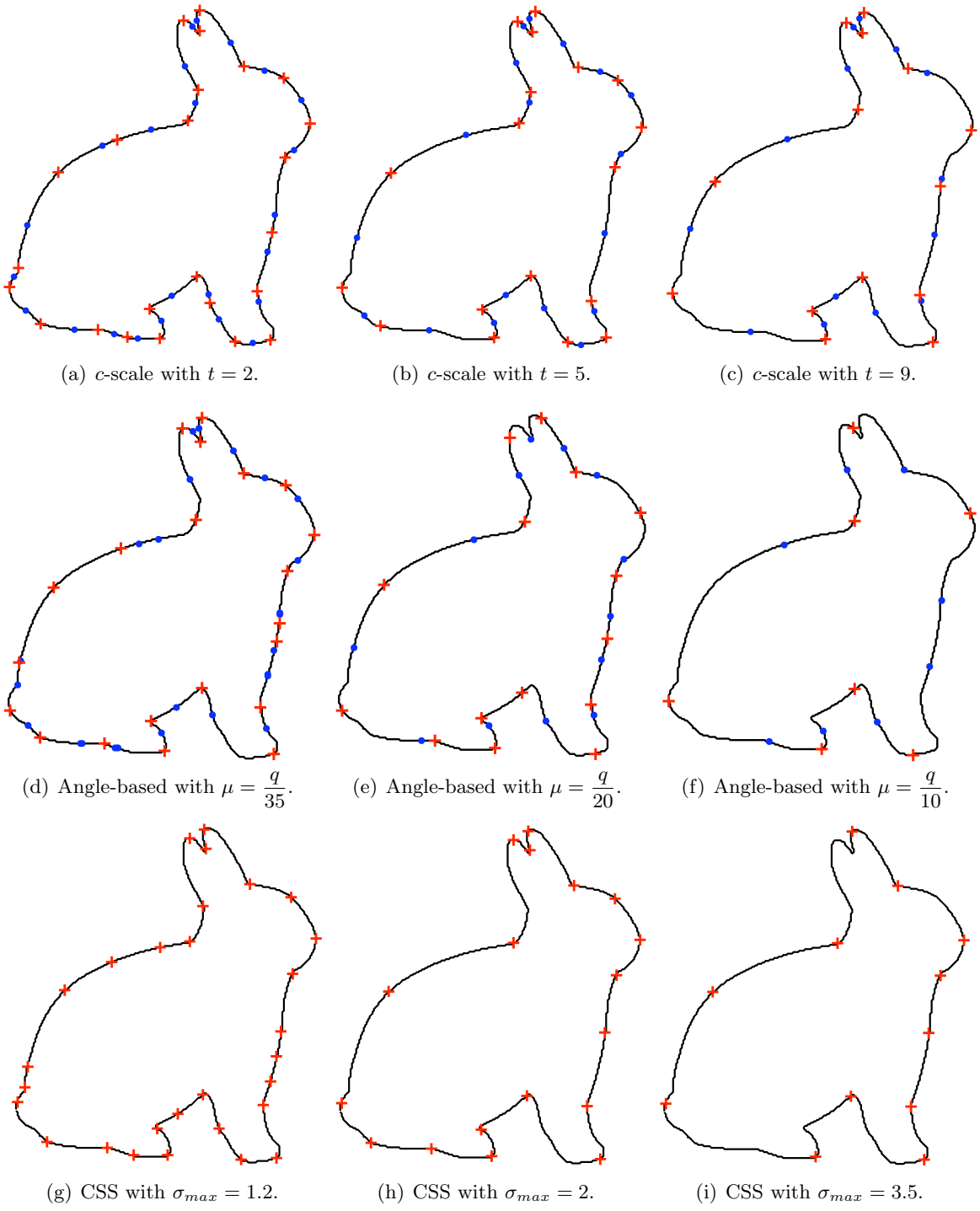
We have illustrated in the previous section the quantitative comparison of shape description accuracy on mathematical and geometrical shapes. In this section, we focus on the extraction of dominant points on more general shapes, where dominant points can be visually identified at a more intuitive level [Attneave, 1954]. We present the results for digital boundaries of natural and medical objects, described in Section 4.1. Since the true dominant points and

their true location cannot be ascertained for these shapes, we qualitatively compare the results obtained for the four methods via image displays.

4.4.1 Non-medical objects

First, we study the influence of the scale or level of detail among methods by taking the rabbit shape as an example. The results are in Figs. 4.27(a)-4.27(c) for the c-scale method, where the c-scale parameter t was changed to see how many dominant points were detected. Similarly, the results for the angle-based method are shown in Figs. 4.27(d)-4.27(f), for different values of v, selected to obtain a similar number of dominant points on the same rabbit shape. The CSS method, by varying the parameter σ_{max}, can identify dominant points at different scales, as shown in Figs.4.27(g)-4.27(i). The ACORD method is not multiscale, as it finds the most appropriate scale to use at each bel, therefore, we cannot show the influence of the scale on that approach and cannot vary the level of detail (a strength and a drawback at the same time).

For the c-scale method, we used $se_v = se_p = 5$ and $t = 2$, $t = 5$, and $t = 9$. As t increases, the number of dominant points detected decreases and the level of detail captured diminishes. At a higher scale (t value, although this is not the scale value σ considered in scale space approaches), most representative corners and inflection points are captured. For the angle-based method, we used $v = 35$, $v = 20$, and $v = 10$. As we decrease the parameter μ (or increase v), we decrease the number of dominant points obtained as well as the accuracy of their placement (see nose and ears of the rabbit in Figs. 4.27(d)-4.27(f)). The location of the landmarks in this method depends on the value selected for μ, whereas in the c-scale method, the position found for the same definitive prominent corners at different scales t is the same. The accuracy does not depend on the scale, therefore the method is more robust to scale changes. We can also notice that, in the angle-based approach, some important corners are not detected and some are misplaced. If we keep on diminishing μ, clusters appear near some of the dominant points detected, leading to repetition of points, and the result is not acceptable anymore for landmark selection. This phenomenon does not



(a) c-scale with $t = 2$. (b) c-scale with $t = 5$. (c) c-scale with $t = 9$.

(d) Angle-based with $\mu = \frac{q}{35}$. (e) Angle-based with $\mu = \frac{q}{20}$. (f) Angle-based with $\mu = \frac{q}{10}$.

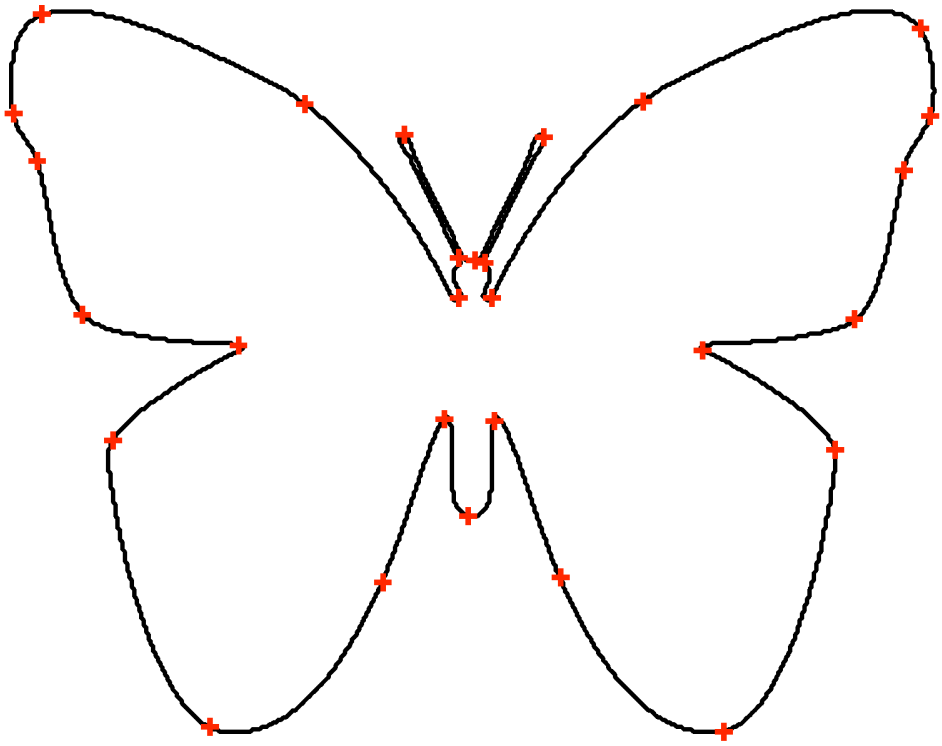
(g) CSS with $\sigma_{max} = 1.2$. (h) CSS with $\sigma_{max} = 2$. (i) CSS with $\sigma_{max} = 3.5$.

Figure 4.27: (a-c) Results of varying the c-scale level of detail parameter t for the rabbit shape in the c-scale method ($w = 3$, $m = 2$, $se_v = se_p = 5$). (d-f) Results of varying the parameter v for the rabbit shape in the angle-based method. (d-f) Results of varying the parameter σ_{max} for the rabbit shape in the CSS method ($\sigma_{min} = 0.1$).

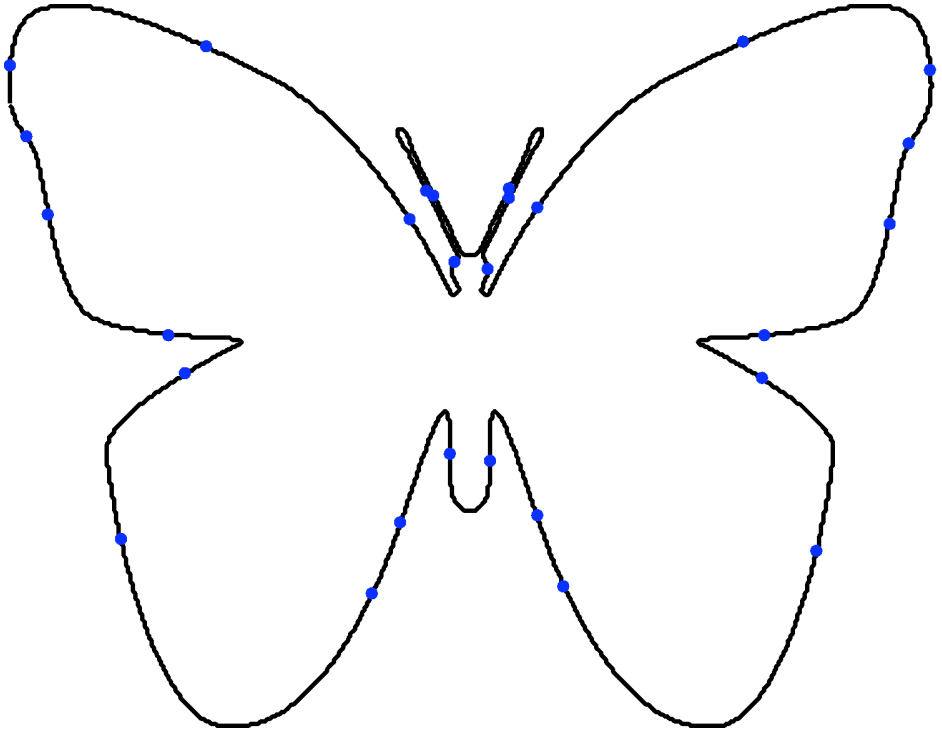
occur in the c-scale method. A dominant point can be detected only once, and it is either a corner or an inflection point. For the CSS approach, we selected $\sigma_{max} = 1.2$, $\sigma_{max} = 2$, and $\sigma_{max} = 3.5$ to obtain scales equivalent to the two previous methods studied. CSS performs similarly to c-scale in the sense that it does not change the position of the same definitive prominent corners detected at different scales. However, it does not seem to retain as well all the prominent points with similar curvature at a higher scale as c-scale does. This can be noticed in the ears on the rabbit in Fig. 4.27(i). Therefore, we may conclude that c-scale is more robust and accurate to scale changes than the angle-based and the CSS methods.

For the butterfly shape (Fig. 4.3(b)), the results from the four methods are displayed in Figs. 4.28-4.30. On this shape, c-scale seems to capture perfectly all high curvature points with enough detail. The angle-based and the CSS approaches are able to distinguish most dominant points on the shape, although they miss some details around the head, on each side of each of the whiskers. The angle-based approach seems a bit more accurate in the detection of the corners than the CSS strategy. The ACORD method captures all the prominent points in the body of the butterfly shape, while it misses more subtle but obvious high curvature points, as the ones defining the shape of the wings. Overall, c-scale performs better than the other three methods. It does not miss prominent corners and it is able to capture the subtle curvature regions around the wings, with higher level of detail than any of the methods studied, in addition to detecting all (some even very subtle) inflection points.

For the fish shape, the results for the four methods are shown in Fig. 4.31. c-scale and CSS seem to perform with higher level of detail than the other two methods. The angle-based approach misses one high curvature point in the boundary, at the base of the tail of the fish (Fig. 4.31(c)). The ACORD method misses certain prominent points and presents some clusters of points around some corners. c-scale seems to capture more subtle curvature points than any of the methods, as can be observed in the tail and in the bottom fin. It also seems to capture more inflection points than the angle-based approach, giving a higher level of detail.

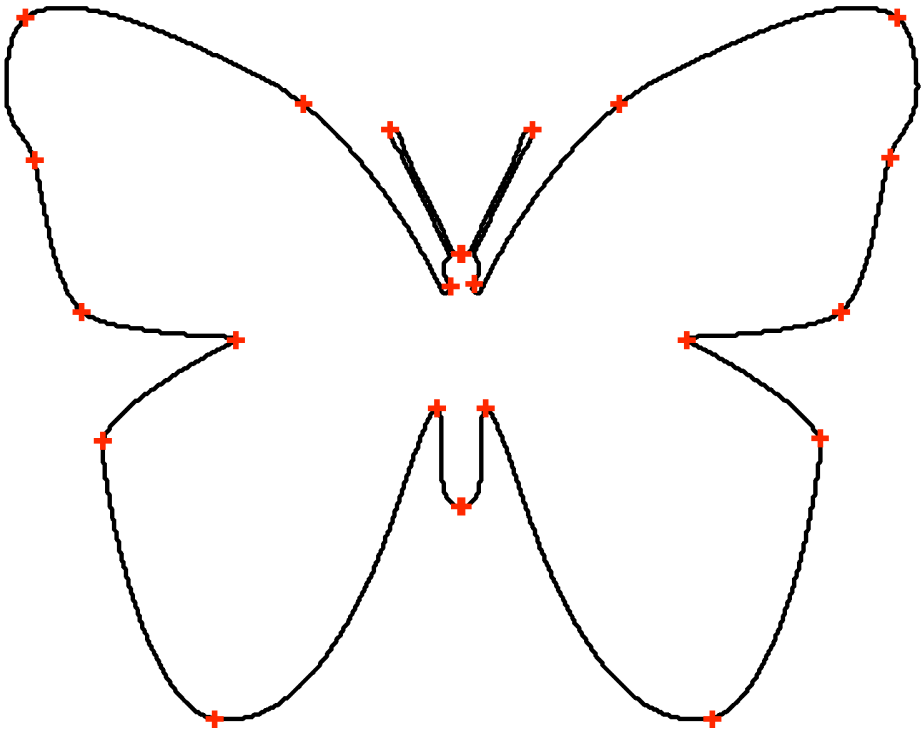


(a) c-scale valleys.

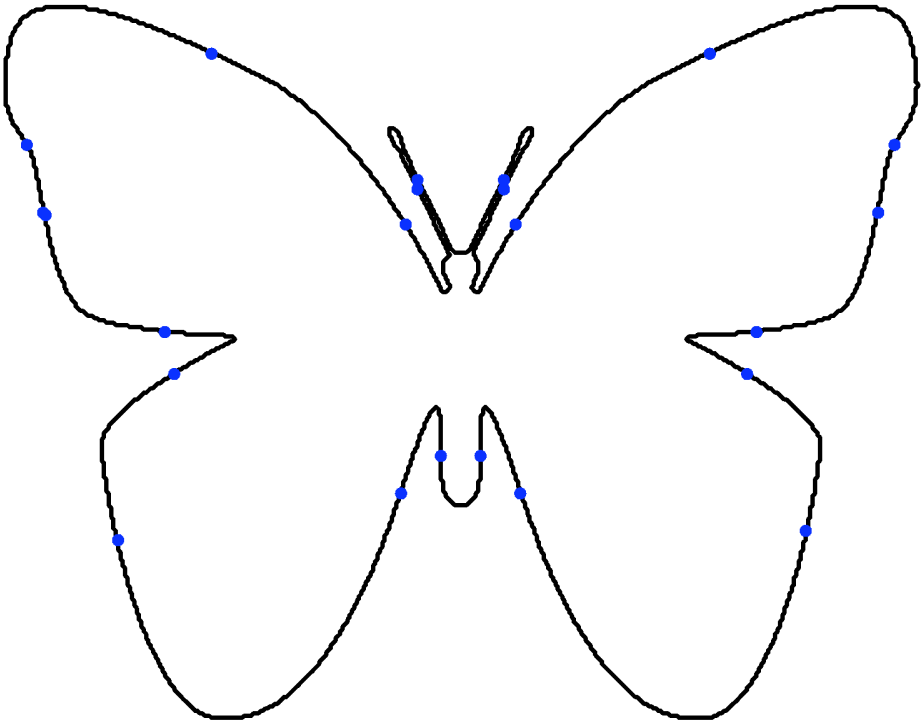


(b) c-scale peaks.

Figure 4.28: Results of applying the c-scale method on a butterfly shape with $t = 2.5$, $w = 3$, $m = 2$, $se_v = se_p = 5$. (a) High curvature points. (b) Inflection points.

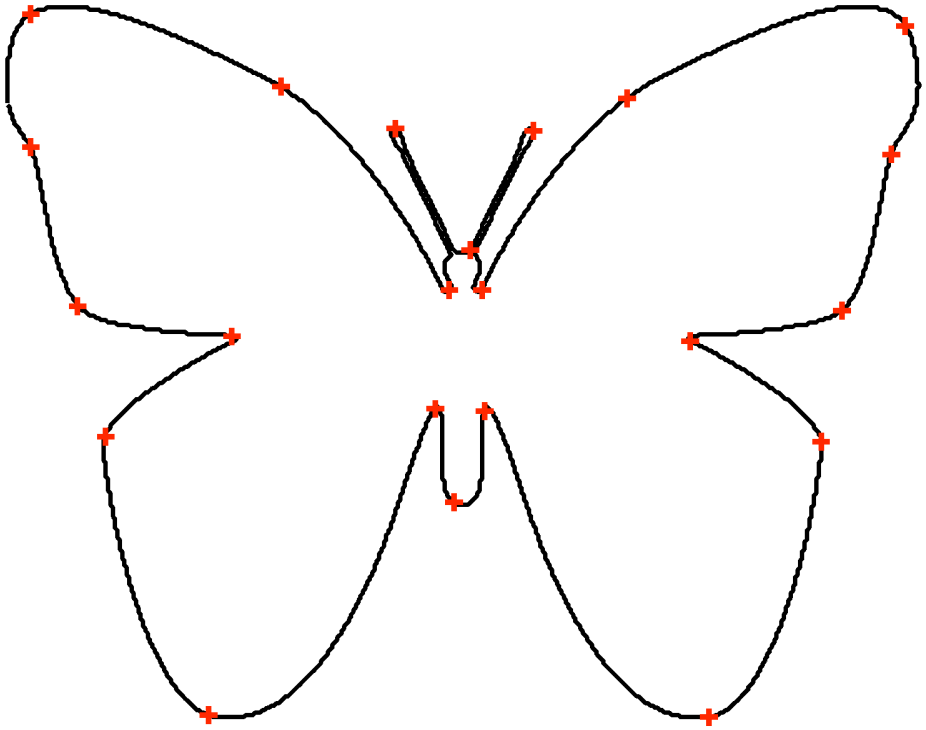


(a) Angle-based maxima.

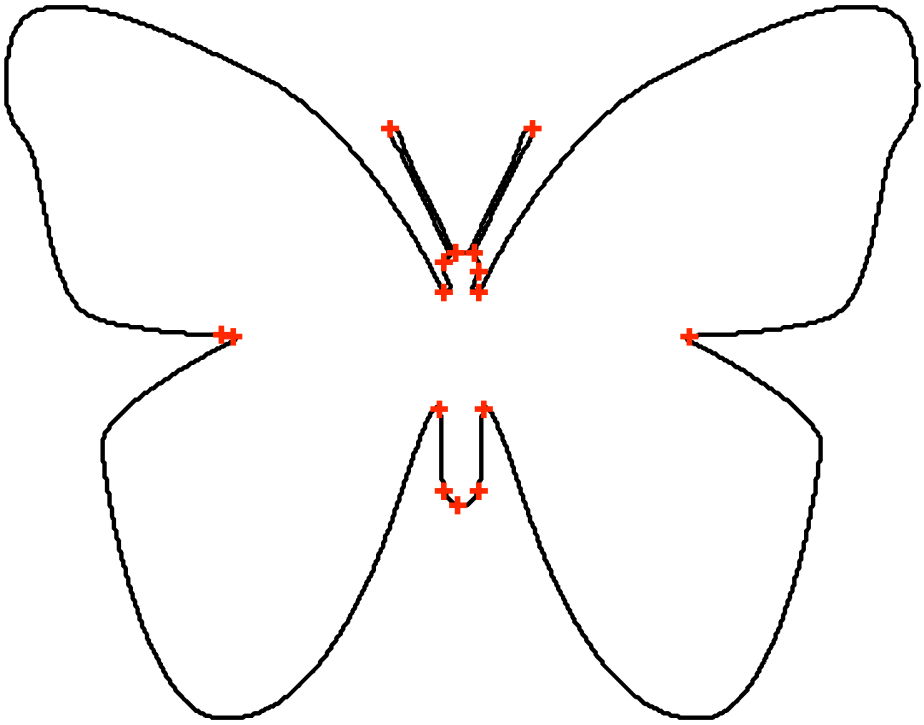


(b) Angle-based minima.

Figure 4.29: Results of applying the angle-based method on a butterfly shape with $v = 50$. (a) High curvature points. (b) Inflection points.

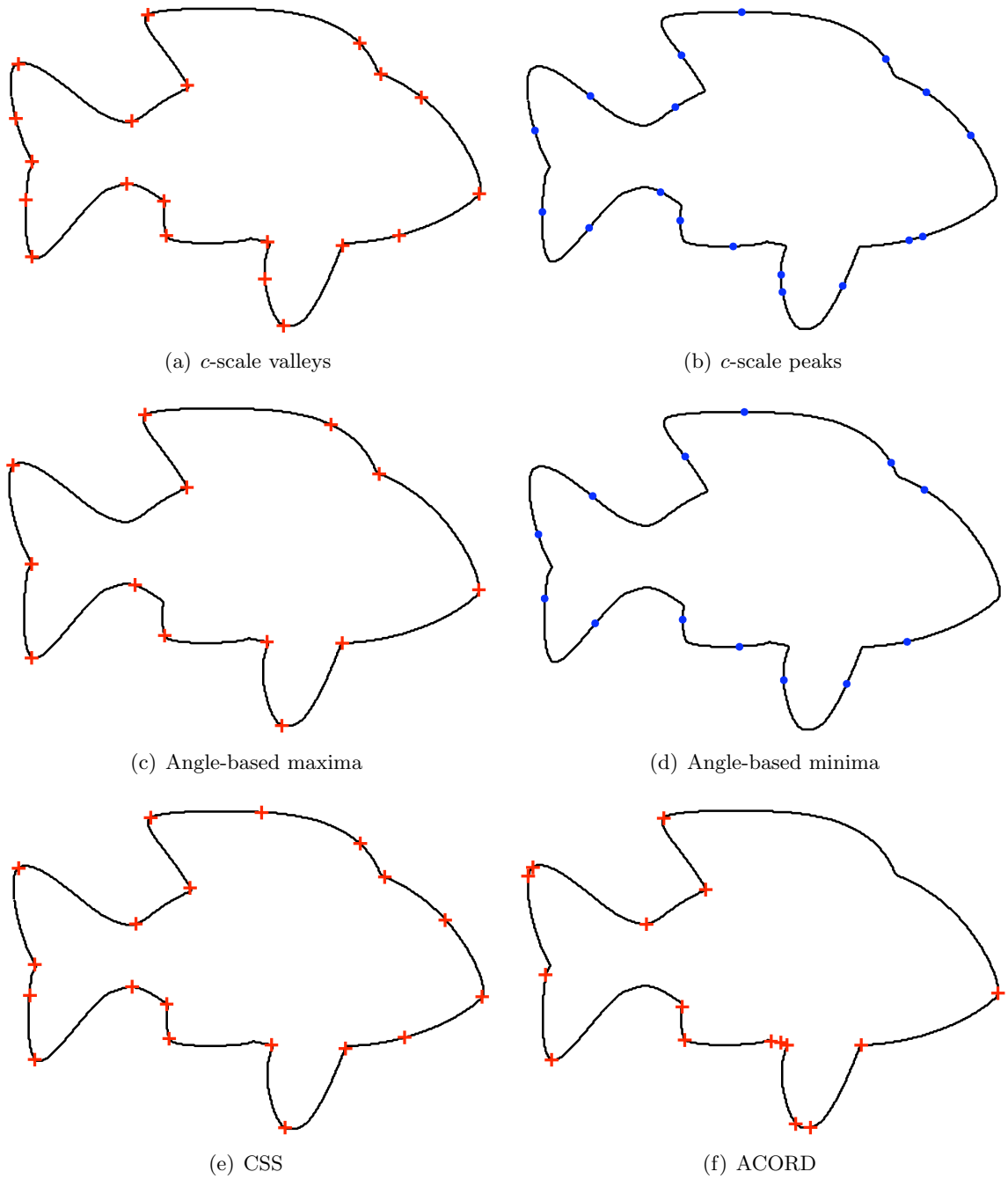


(a) CSS



(b) ACORD

Figure 4.30: Results of applying (a) the CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$, and (b) the ACORD method on a butterfly shape with $T = 0.1$.



(a) c-scale valleys (b) c-scale peaks

(c) Angle-based maxima (d) Angle-based minima

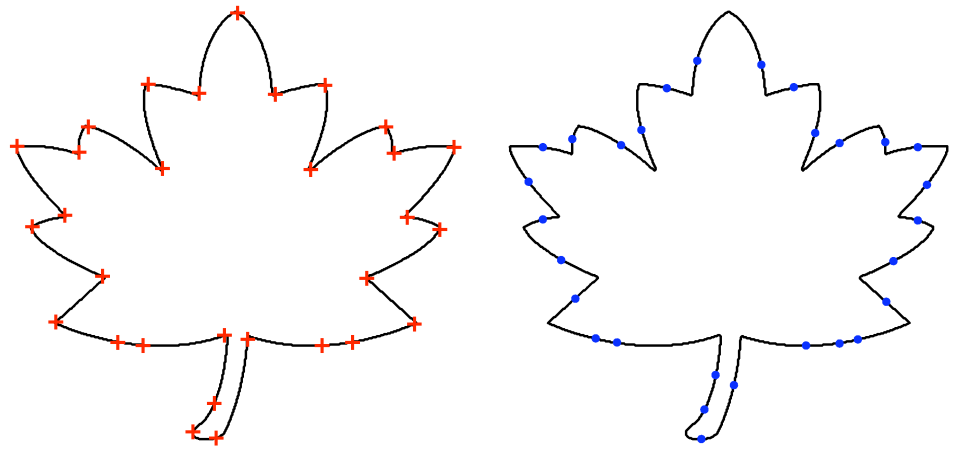
(e) CSS (f) ACORD

Figure 4.31: The dominant points detected on the fish shape. (a) c-scale method with $t = 3.8$, $w = 3$, $m = 2$, $se_v = se_p = 10$. (b) Angle-based method with $v = 24$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 2$. (d) ACORD with $T = 1$.

For the leaf shape, c-scale and CSS perform similarly and with higher level of detail than any other method, although c-scale finds a corner at the bottom of the stem that the CSS approach misses. The angle-based approach is unable to capture the details at the bottom of the stem, and misses two prominent corners on the leaf. It also misses some inflection points in some parts of the leaf and misplaces other inflection points locating them at corner positions. The ACORD method detects most of the prominent points, but is unable to detect corners at more subtle curvature regions, such as the bottom of the leaf and the curvature present in the stem. It also finds a FP next to one of the detected corners on the top portion. c-scale detects all and only the intuitive dominant points existing in the leaf shape.

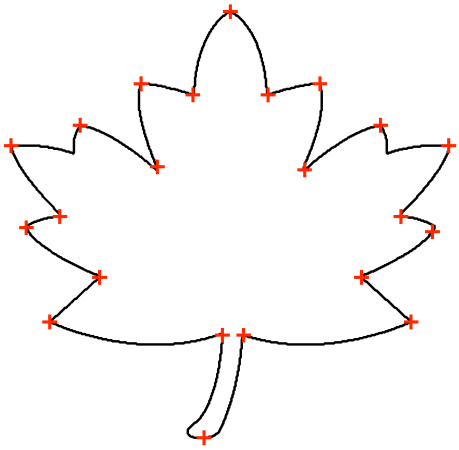
For the hand shape, the angle-based and the CSS approaches obtain very similar results. The ACORD method finds many corners on the tips of the fingers, while it is also unable to capture softer curvature points at the bottom of the hand or middle part of the fingers. c-scale describes the shape with a high level of detail, as we can notice in the fingers and palm, finding all corners from the most prominent to the most subtle changes of curvature. It also captures the corresponding inflection points with more detail than the angle-based approach.

For the deer shape, the angle-based approach finds most of the corners except for some prominent corners existing at the bottom of the paws. It also presents some inaccuracies in the localization of certain corners such as the tip of the tail. CSS detects most of the corners existing in the shape, except for some at the bottom of the paws. The ACORD method finds only the most prominent corners in the shape, but misses all subtle changes of curvature, like those present at the bottom (and upper) part of the deer's body. It also detects clusters of corners around certain regions, such as the frontal leg. c-scale captures all corners with different degrees of curvature. It also finds correctly all inflection points that may be intuitively located in the shape, whereas the angle-based method presents clusters of points for some of the inflection points detected.

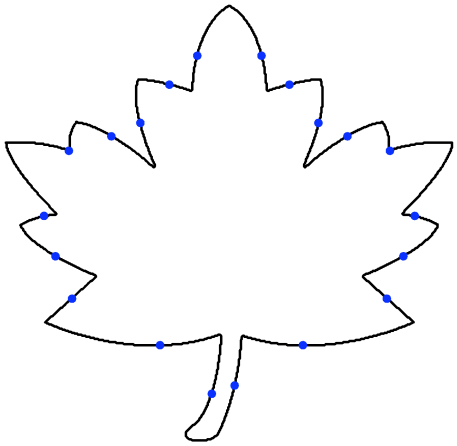


(a) c-scale valleys

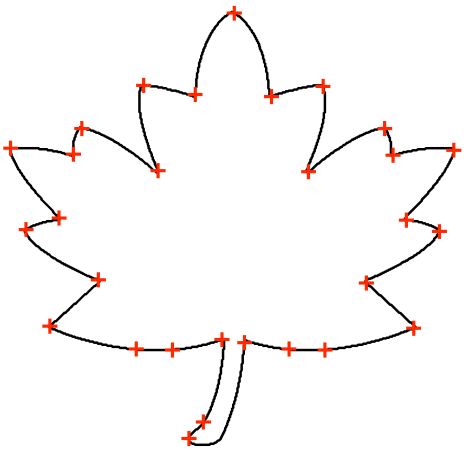
(b) c-scale peaks



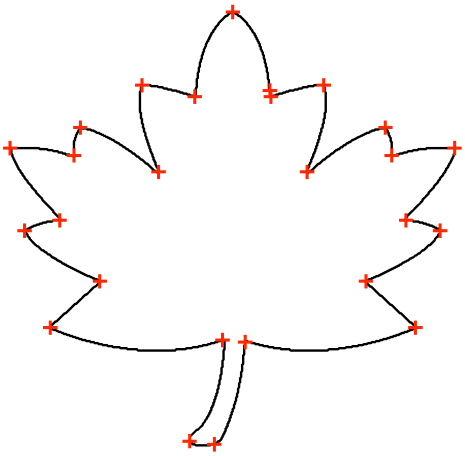
(c) Angle-based maxima



(d) Angle-based minima

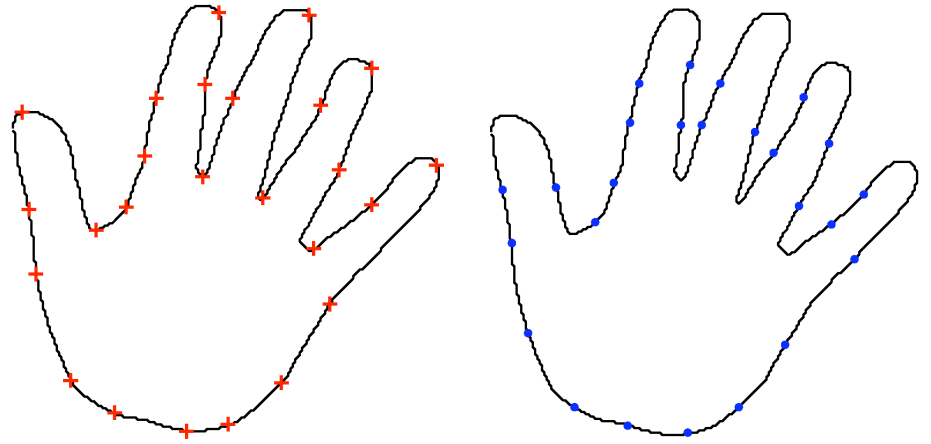


(e) CSS



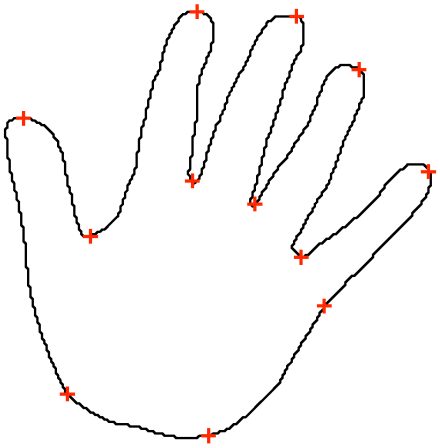
(f) ACORD

Figure 4.32: The dominant points detected on the leaf shape. (a) c-scale method with $t = 5$, $w = 3$, $m = 2$, $se_v = se_p = 5$. (b) Angle-based method with $v = 23$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (d) ACORD with $T = 0.1$.

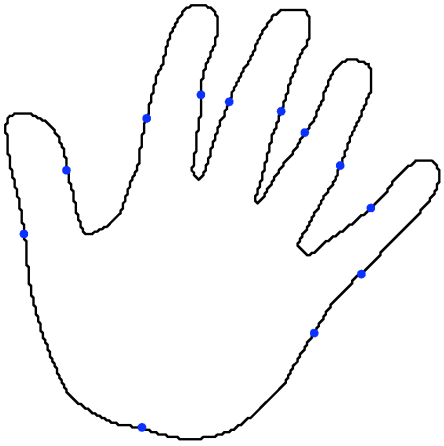


(a) c-scale valleys

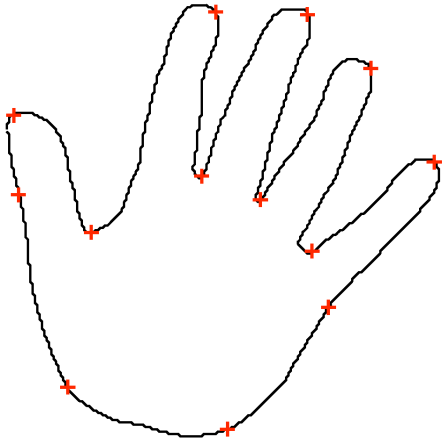
(b) c-scale peaks



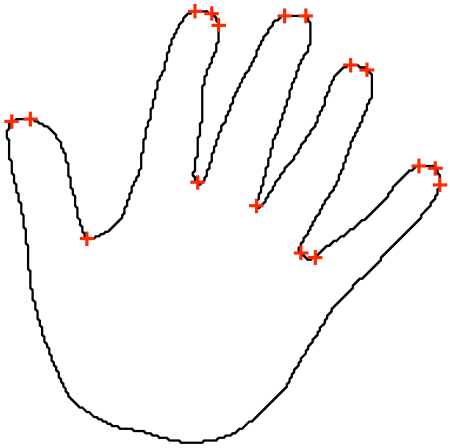
(c) Angle-based maxima



(d) Angle-based minima

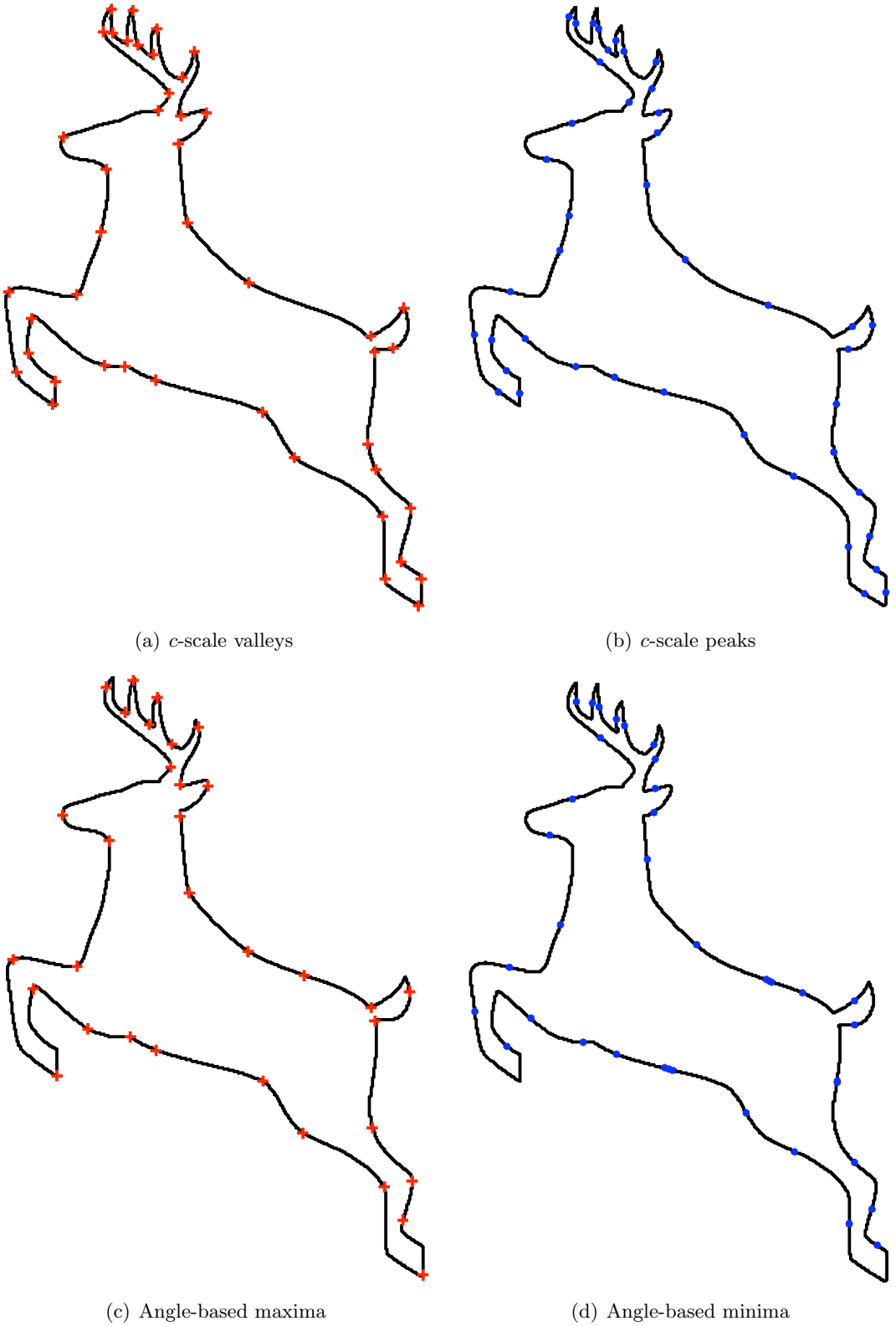


(e) CSS



(f) ACORD

Figure 4.33: The dominant points detected on the hand shape. (a) c-scale method with $t = 1.5$, $w = 3$, $m = 2$, $se_v = se_p = 5$. (b) Angle-based method with $v = 20$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 2.5$. (d) ACORD method with $T = 0.1$.



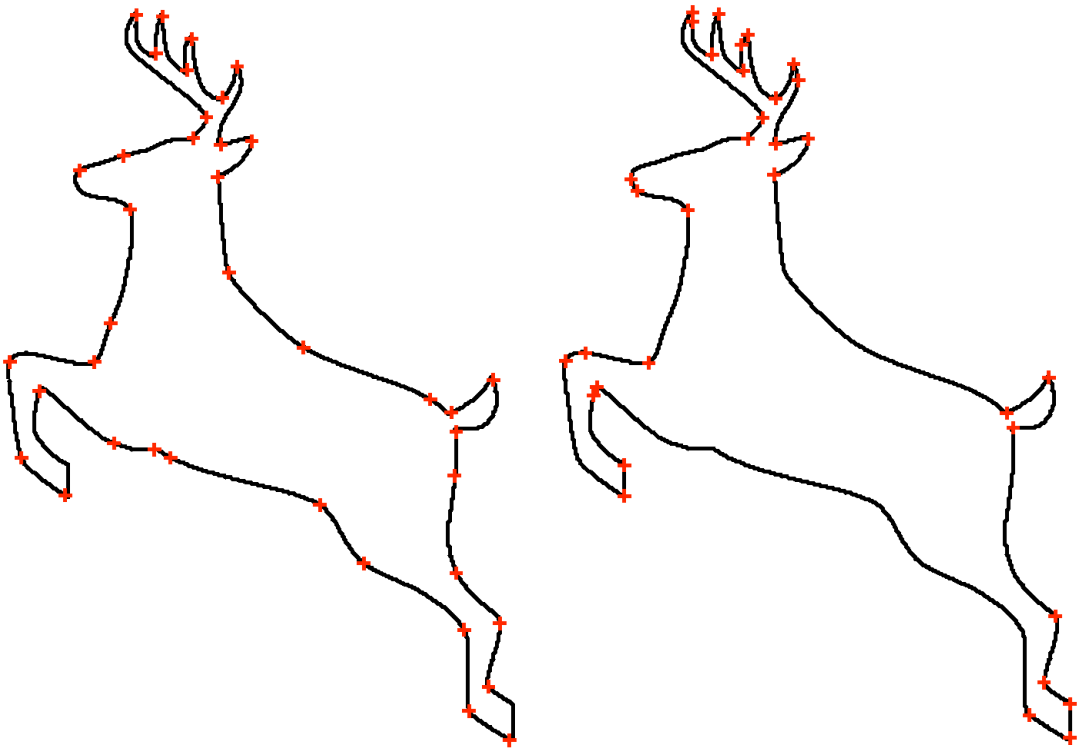
(a) c-scale valleys

(b) c-scale peaks

(c) Angle-based maxima

(d) Angle-based minima

Figure 4.34: The dominant points detected on the deer shape. (a) c-scale method with $t = 2.1$, $w = 3$, $m = 2$, $se_v = se_p = 5$. (b) Angle-based method with $v = 50$.



(a) CSS (b) ACORD

Figure 4.35: The dominant points detected on the deer shape. (a) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 2$. (b) ACORD with $T = 0.1$.

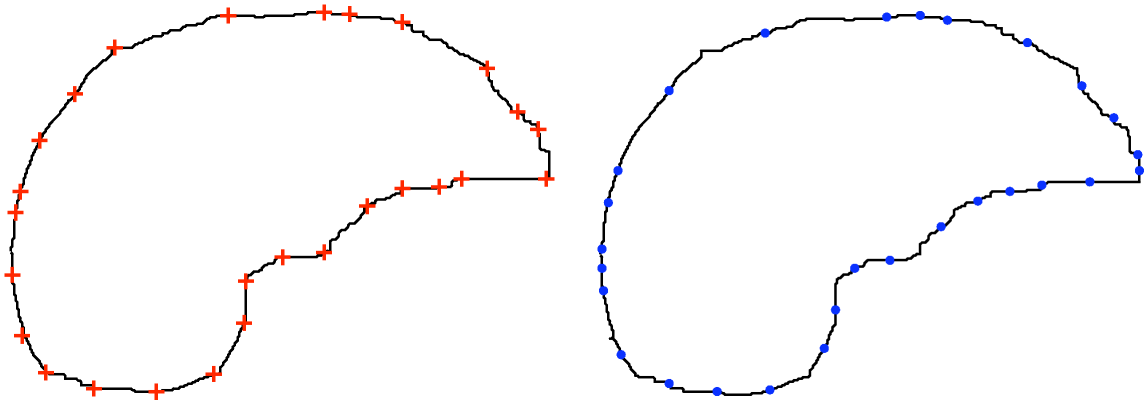
Overall, from the above qualitative analysis, we conclude that c-scale performs better, more accurately, and with a higher level of detail than the other three methods. It does not miss prominent points or add clusters of points around certain corners as some of the other methods do while being able to detect very subtle changes for both corners and inflection points. The ACORD method seems to miss most of the subtle changes in curvature and finds some FPs. The CSS method performs most of the time similarly to the c-scale method, but misses certain corners or subtle changes of curvature in some cases. The angle-based approach is less accurate than the other methods. It is also highly sensitive to the selection of the parameter v, whereas the other methods are more robust to parameter changes.

4.4.2 Medical objects

The ability of the methods to detect dominant points is qualitatively illustrated for the six medical objects in Figs. 4.36-4.43. Since the true corners (and inflection points) and their locations are not known for these shapes, only a display of the points reported is presented.

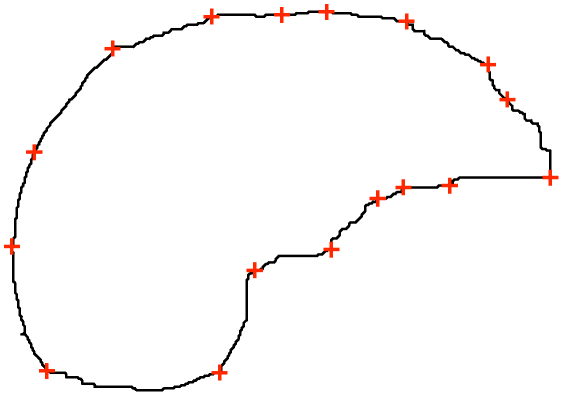
Consistent with the analysis up to this point, the c-scale approach seems to perform better than other methods and is able to capture accurately the details of the shapes even with a small number of landmarks, as illustrated in Figs. 4.36(a), 4.37(a), 4.38(a), 4.39(a), 4.40(a), and 4.41(a). The angle-based and the CSS methods seem to miss certain corners (Figs. 4.37(c), 4.37(e), 4.38(c)) and detect some points that are likely to be noise (Figs. 4.40(c), 4.39(c)). Moreover, if we observe the cortical sulci closely, the dominant points identified by the CSS method (Fig. 4.43(a)) are not exactly located in the maxima of curvature in that region, but seem to be shifted from the centre of the sulci. This can also be appreciated for the calcaneus shape in Fig. 4.38(e). The c-scale approach (Figs. 4.41 and 4.38(a)), however, seems to locate the points consistently accurately. The ACORD method performs well on the cortex (Fig. 4.43(b)) and seems to find corners accurately and with enough detail for that shape. However, for all other shapes, it seems to detect false/noisy corners, such as in the ventricle (Fig. 4.40(f)), and for most cases it does not find corners at subtle curvature regions (Fig. 4.37(f)), while producing clusters of points around certain detected corners, as can be noticed in the ventricle and vertebra shapes.

The inflection points detected by the angle-based method have for certain shapes clusters of points in certain regions, as can be observed in Figs. 4.38(d), 4.39(d), and 4.40(e). c-scale performs better on the inflection points than the angle-based approach as it does not present any cluster of points, as demonstrated in Figs. 4.36(b), 4.37(b), 4.38(b), 4.39(b), 4.40(b), and 4.41(b).

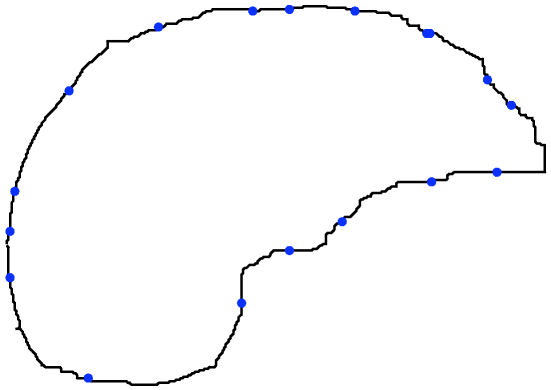


(a) c-scale valleys

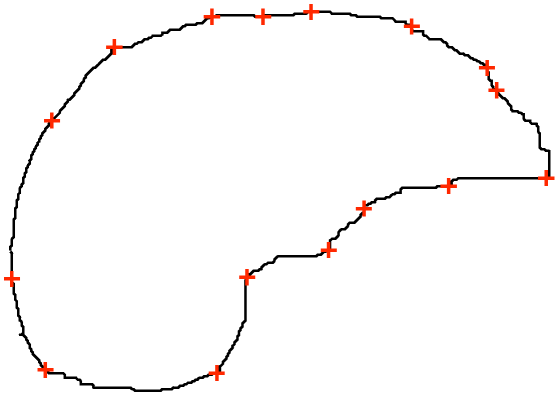
(b) c-scale peaks



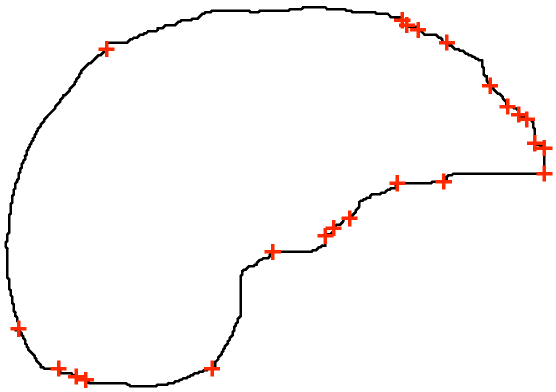
(c) Angle-based maxima



(d) Angle-based minima

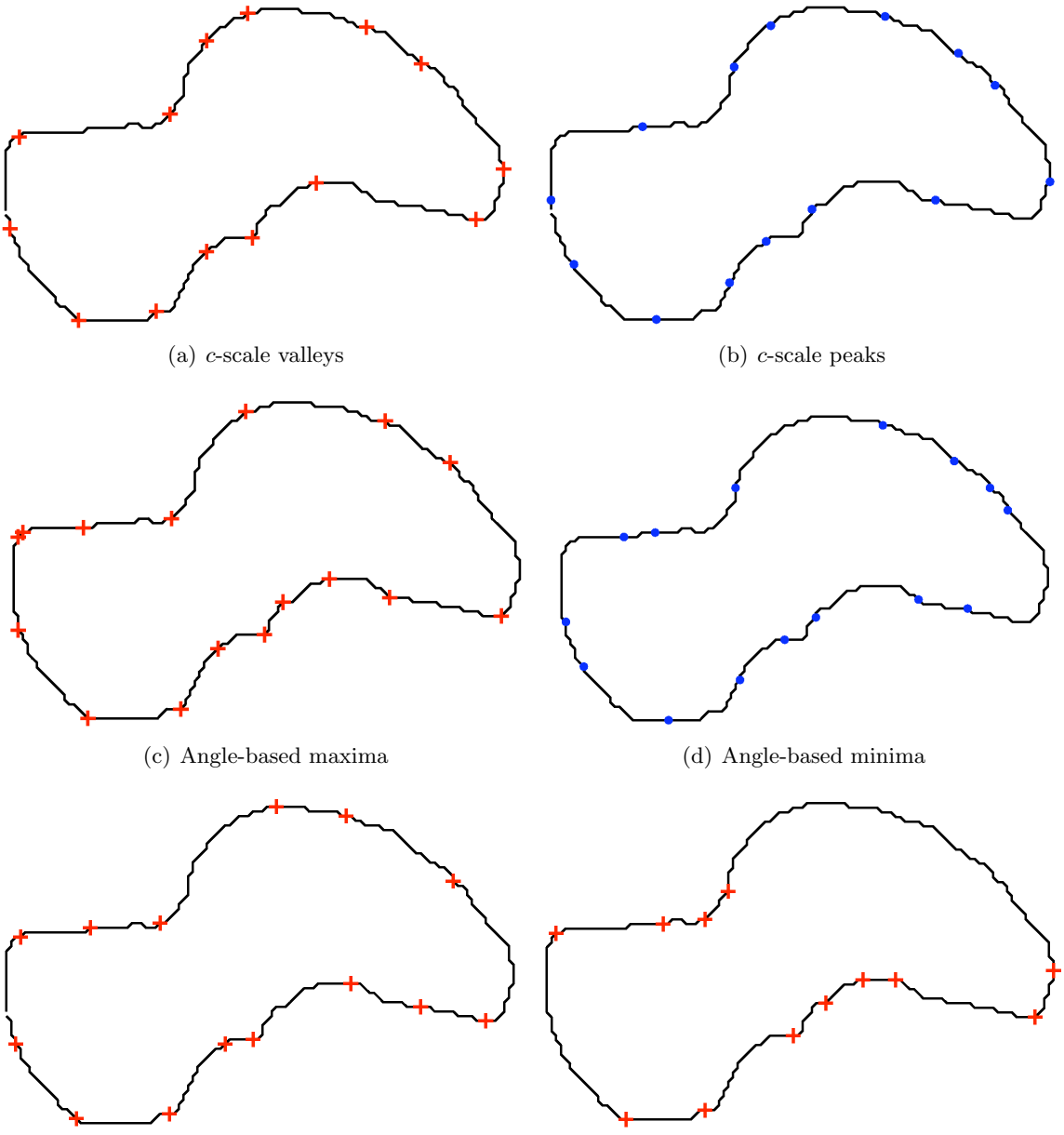


(e) CSS



(f) ACORD

Figure 4.36: (a-b) c-scale valley and peak detection on liver with $t = 3.2$, $w = 3$, $m = 3$, $se_v = se_p = 15$. (c-d) Angle-based method on liver with $v = 27$. (e) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (f) ACORD with $T = 0.1$.

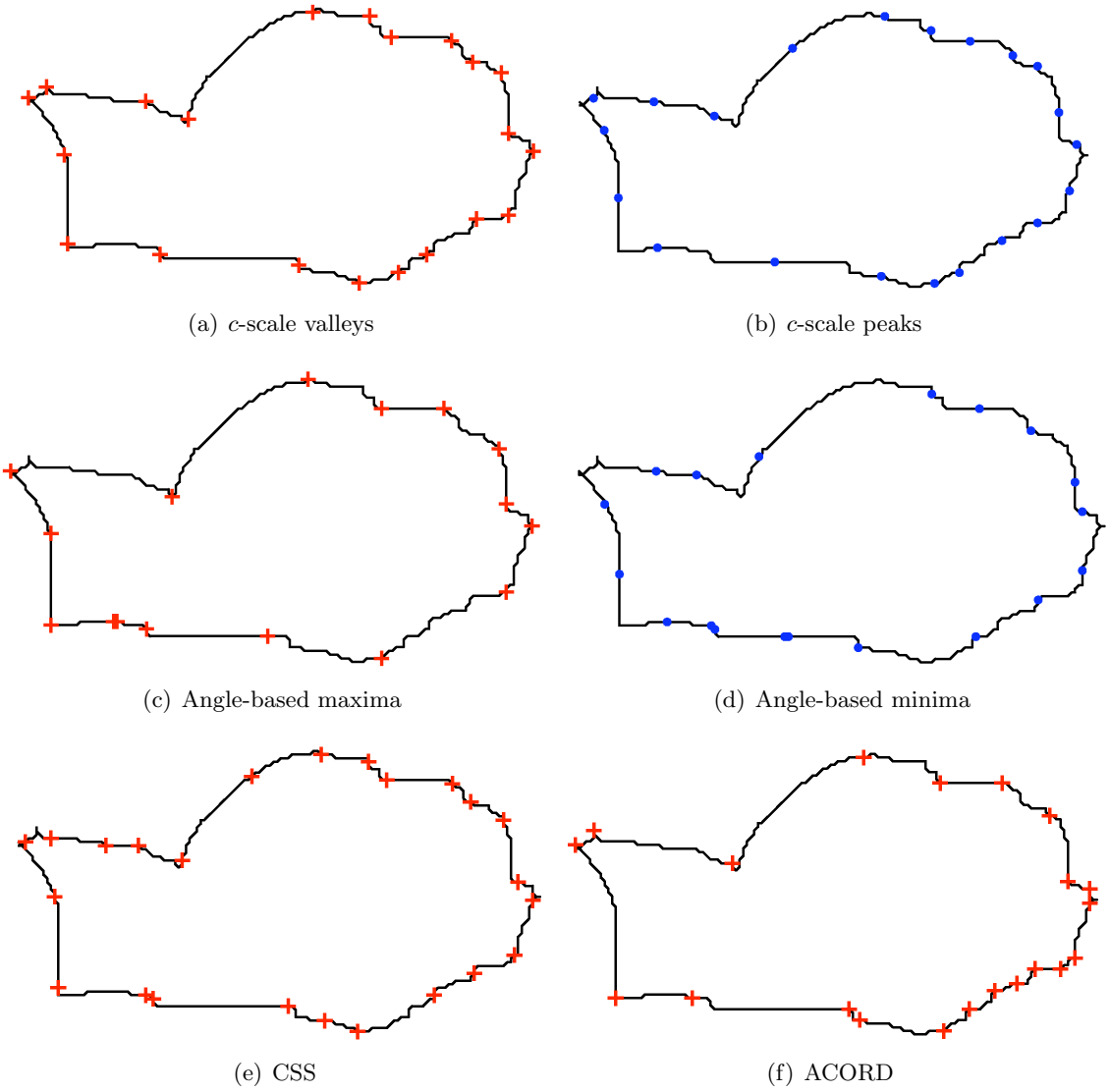


(a) c-scale valleys (b) c-scale peaks

(c) Angle-based maxima (d) Angle-based minima

(e) CSS (f) ACORD

Figure 4.37: (a-b) c-scale valley and peak detection on talus with $t = 3.1$, $w = 2$, $m = 2$, $se_v = se_p = 5$. (c-d) Angle-based method on talus with $v = 20$. (e) CSS method with $\sigma_{min} = 0.01$ and $\sigma_{max} = 1$. (f) ACORD with $T = 0.2$.

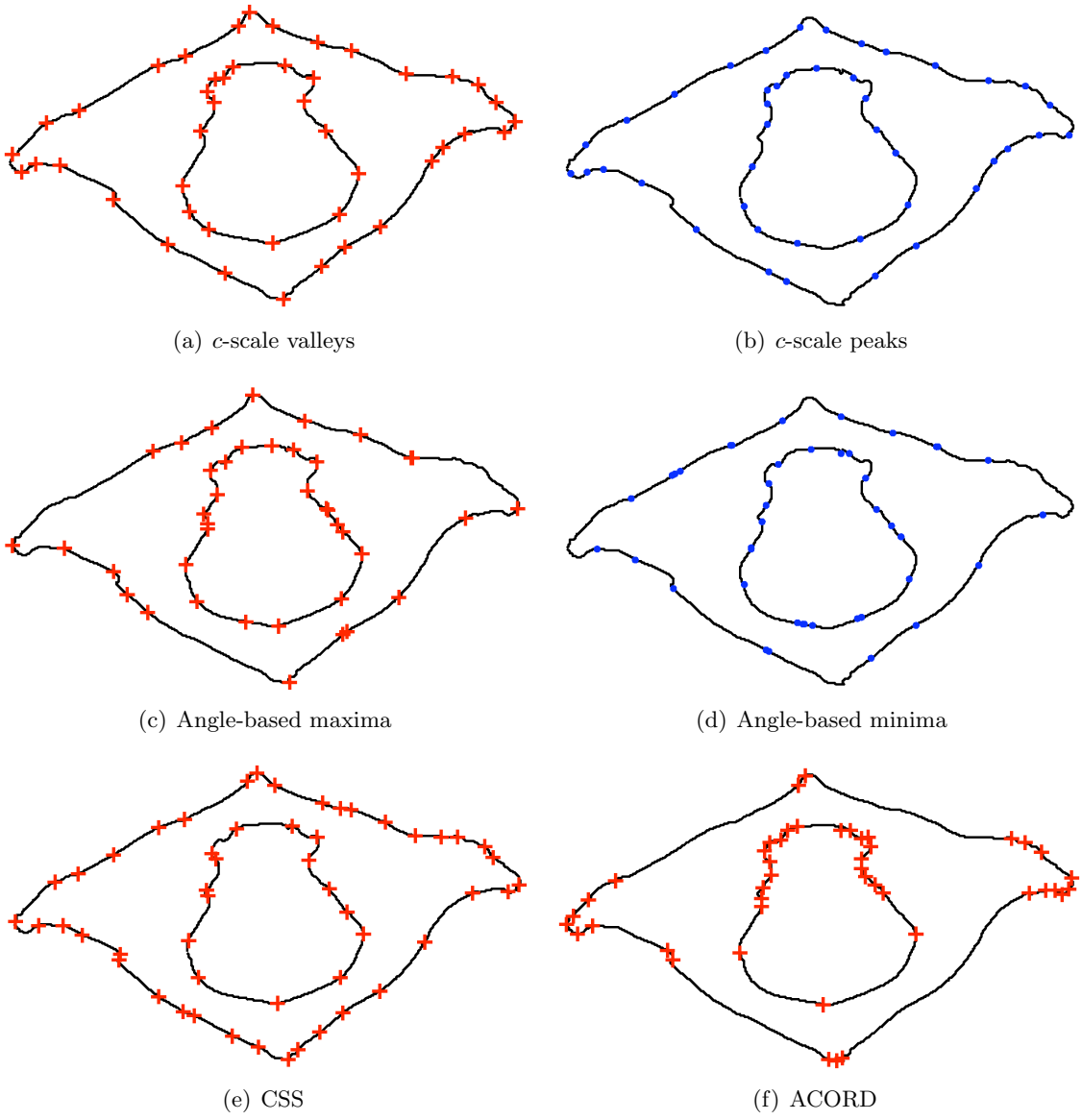


(a) c-scale valleys (b) c-scale peaks

(c) Angle-based maxima (d) Angle-based minima

(e) CSS (f) ACORD

Figure 4.38: (a-b) c-scale valley and peak detection on calcaneus with $t = 1.5$, $w = 2$, $m = 2$, $se_v = se_p = 5$. (c-d) Angle-based method on calcaneus with $v = 20$. (e) CSS method with $\sigma_{min} = 0.01$ and $\sigma_{max} = 1$. (f) ACORD with $T = 0.1$.



(a) c-scale valleys

(b) c-scale peaks

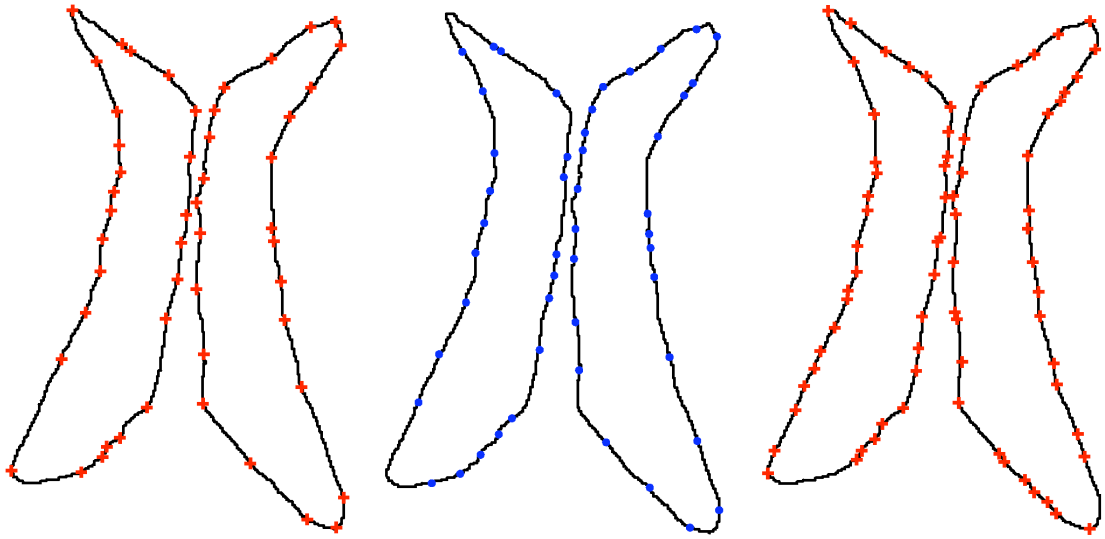
(c) Angle-based maxima

(d) Angle-based minima

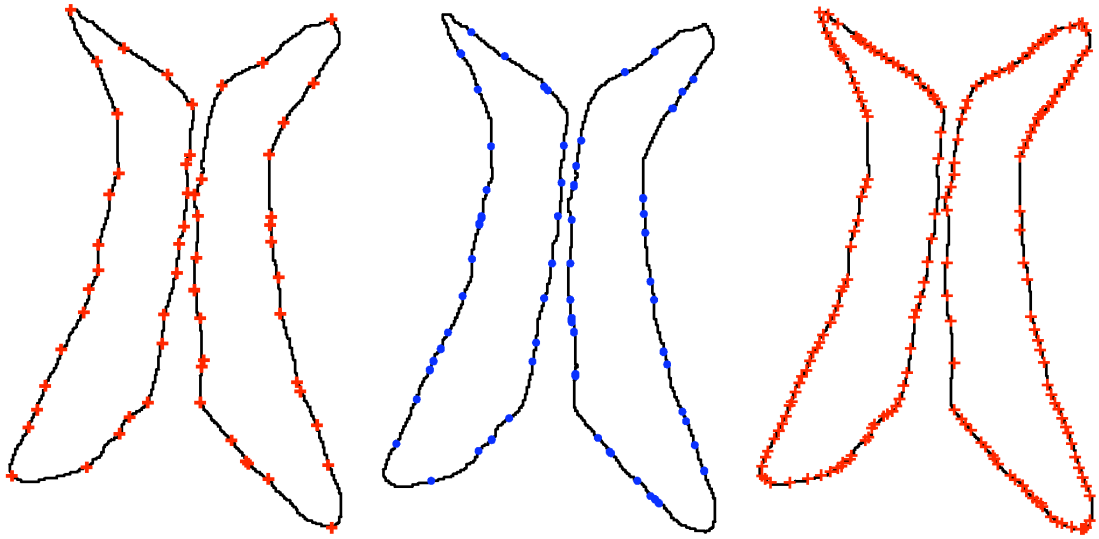
(e) CSS

(f) ACORD

Figure 4.39: (a-b) c-scale valley and peak detection on vertebra with $t = 2.1$, $w = 2$, $m = 2$, $se_v = se_p = 5$. (c-d) Angle-based method on a vertebra with $v = 25$. (e) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 1.3$. (f) ACORD with $T = 0.08$.

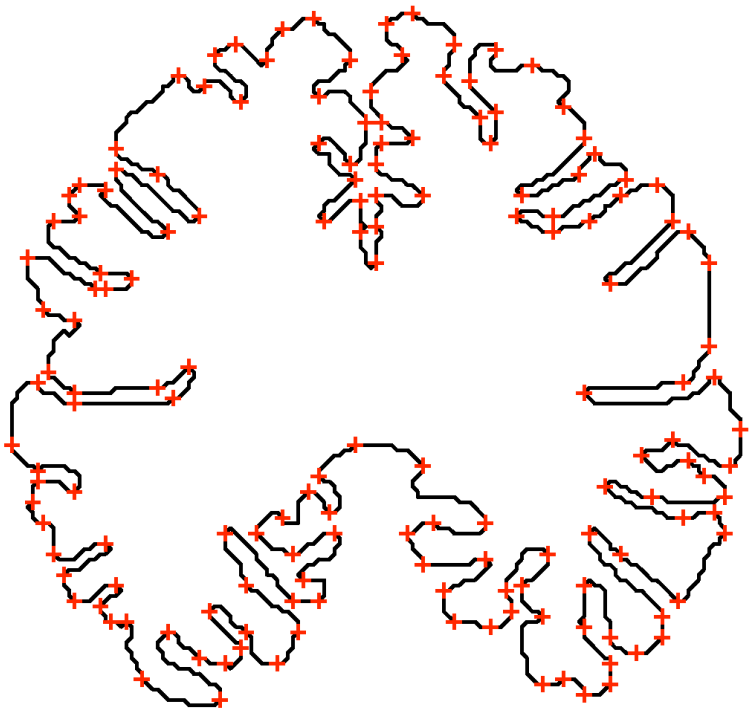


(a) c-scale valleys (b) c-scale peaks (c) CSS

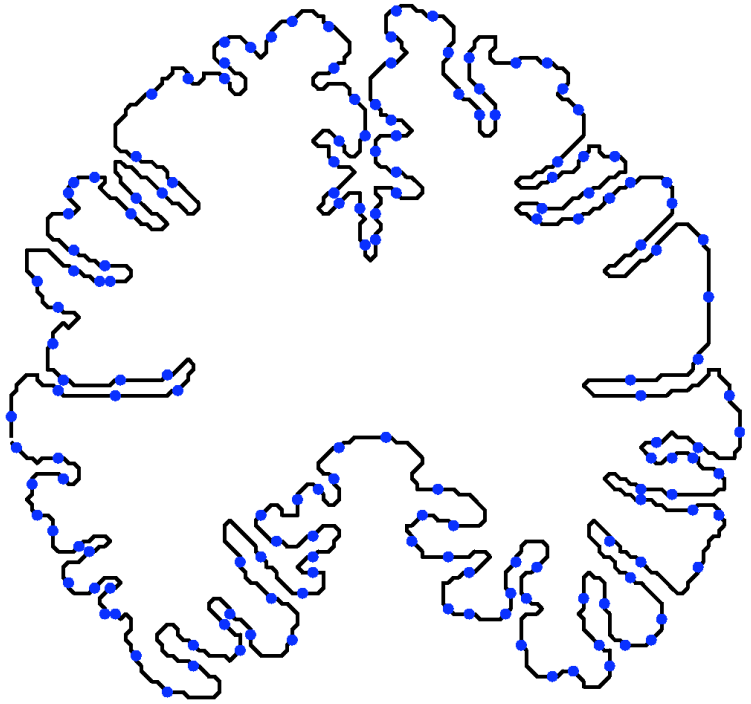


(d) Angle-based maxima (e) Angle-based minima (f) ACORD

Figure 4.40: (a-b) c-scale valley and peak detection on the brain ventricles shape with $t = 5$, $w = 4$, $m = 3$, $se_v = se_p = 25$. (c) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 3$. (d-e) Angle-based method on a vertebra with $v = 36$. (f) ACORD with $T = 0.35$.

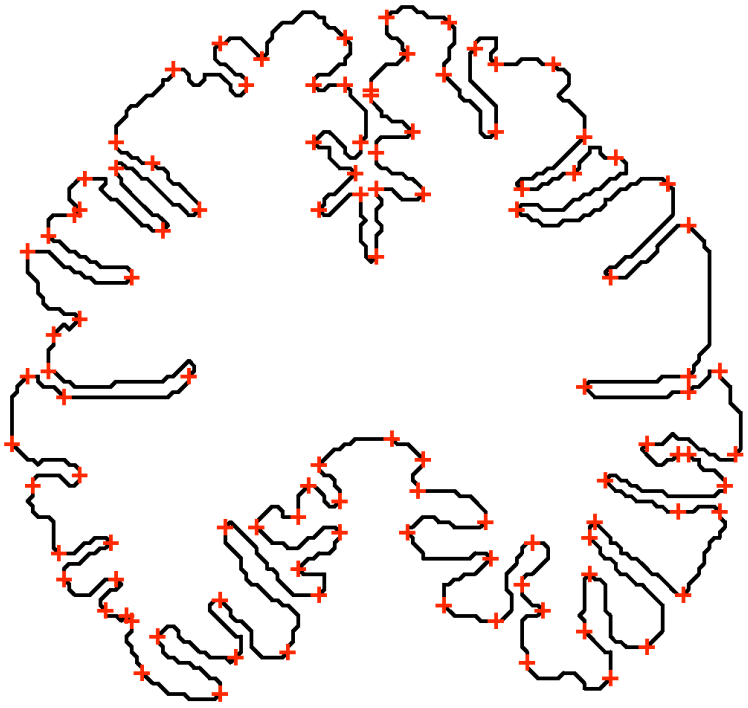


(a) c-scale valleys

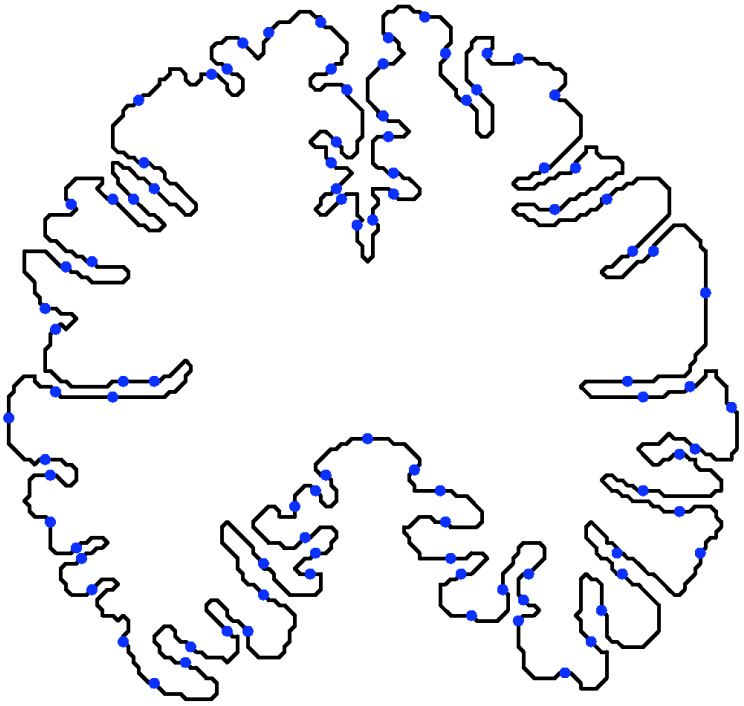


(b) c-scale peaks

Figure 4.41: The dominant points detected on the brain cortex shape with the c-scale method ($t = 1$, $w = 1$, $m = 1$, $se_v = se_p = 3$). (a) High curvature points. (b) Inflection points.

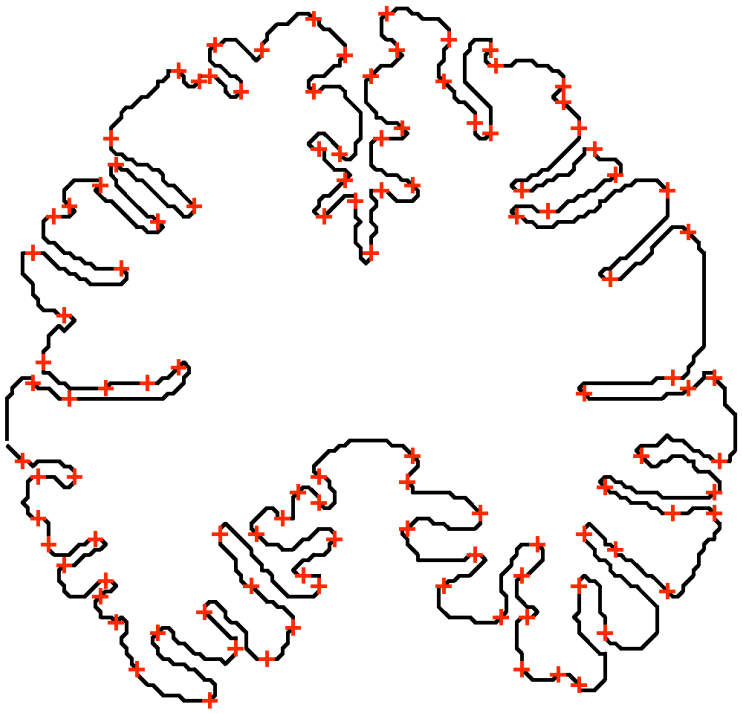


(a) Angle-based maxima

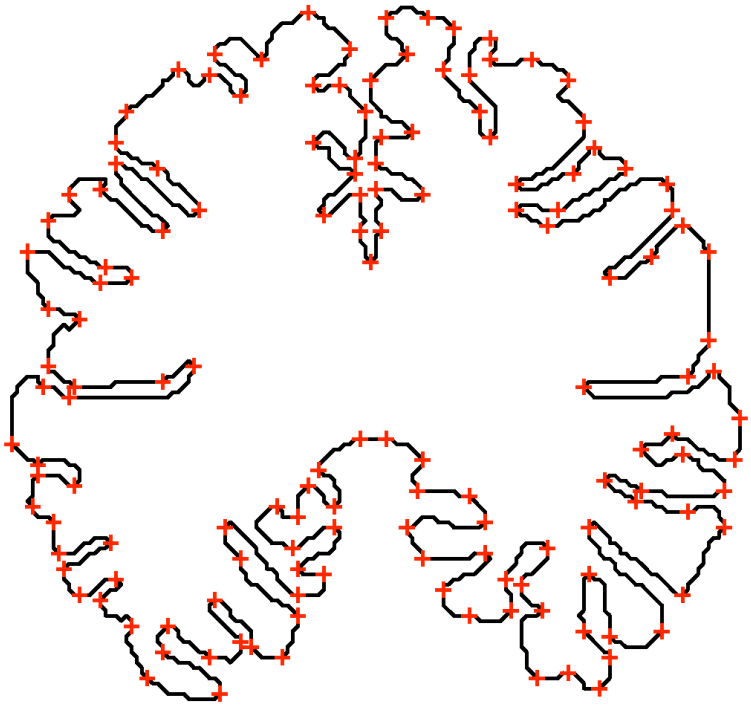


(b) Angle-based minima

Figure 4.42: The dominant points detected on the brain cortex shape with the angle-based method ($v = 130$).



(a) CSS



(b) ACORD

Figure 4.43: The dominant points detected on the brain cortex shape. (a) CSS method with $\sigma_{min} = 0.1$ and $\sigma_{max} = 0.6$. (b) ACORD with $T = 0.1$.

4.5 Conclusions

In this chapter, we compared global and local scale approaches to shape analysis. For global scale, the Curvature Scale Space (CSS) [Rattarangsi and Chin, 1992] method was selected, which is a state-of-the-art shape descriptor, and is used in the MPEG-7 standard. We also used the adaptive corner detector [Ray and Pandyan, 2003] based on CSS as another global method to compare with our locally adaptive morphometric shape descriptor. The locally adaptive scale approach is based on the notion of curvature-scale (c-scale), which is a new local scale concept that brings the idea of local morphometric scales (such as ball-, tensor-, and generalized scale) developed for images to the realm of boundaries.

Although the idea underlining ACORD has a spirit similar to c-scale, the two approaches are fundamentally different in many respects. Inspired by the CSS idea, ACORD goes a step further in determining the scale that is most appropriate for each point on the boundary. In this sense, its approach is scale adaptive and local. Its main focus is on estimating the curvature at each point on the shape boundary as reliably as possible. In contrast, the goal and the approach of c-scale is broader - one of constructing a rich shape description. One of the byproducts of this description happens to be the ability to detect "corners" or high curvature points. The c-scale method describes any shape (digital or analog) in terms of fundamental units called c-scale segments, which are symmetric, homogeneous, oriented, curve segments. (The motivation for c-scale came from ball-scale for 3D images wherein the image is described in terms of the largest homogeneous ball situated at every voxel in the image under an appropriate homogeneity criterion.) Once this fundamental and simple c-scale description is given, several simple operations can be carried out by using the representation to derive useful shape properties in a robust and reliable manner.

We compared the c-scale method to another local scale approach, which is angle-based [Rosenfeld and Johnston, 1973] and which finds also the inflection points on shapes. In this chapter, we presented a thorough evaluation of these global and local scale methods based

on 21 different kinds of natural and artificial shapes. Our analysis indicates the following:

1. Overall, the c-scale method performed the best among all methods based on accuracy in detection (FPs and FNs) and localization accuracy (% RMS distance). It was the only method with perfect detection accuracy. Its localization accuracy was consistent across all shapes demonstrating non dependence on shape in performance. The second best ACORD method demonstrated similar consistency in quantitative evaluation. However, its performance on some of the natural shapes (medical objects) was quite poor.

2. A useful feature of the c-scale, angle-based, and CSS methods is the ability to control level of detail. ACORD does not offer this feature. c-scale's robustness to scale changes - meaning the consistency and accuracy with which the same dominant points are selected - is quite remarkable and is better than the angle-based and CSS methods.

3. The c-scale and the angle-based methods are capable of detecting inflection points unlike CSS and ACCORD methods. The accuracy of detection and localization of such points by c-scale is considerably superior to that of the angle-based algorithm.

4. c-scale is an adaptation of the ball-scale idea [Saha et al., 2000] for nD images to the 1D boundaries of 2D shapes. It produces a richer description of shape than the other methods compared.

5. The c-scale method's accuracy seems to be consistent across all aspects of shape features such as small or large in size and sharp or subtle in definition. To clarify this point, the angle-based method seems to perform well on sharp angles in large size features. However, its performance deteriorates when sharp angles are encountered in small boundary segments. Similarly, ACORD seems to prefer large segments and sharper definitions and its performance on subtle and small features seems to be poorer.

6. c-scale is a simpler method with no special purpose mechanisms, its implementation is simple, it is defined clearly, and has a lower computational cost than any of the other methods used for comparison purposes.

5

Curvature Scale-Based Landmark Tagging Methods

In this chapter, we present a framework containing several strategies for automatic landmark tagging based on the c-scale shape descriptor introduced in Chapter 3. The main idea consists of automatically characterizing a training set of shapes by shape-salient points selected on their digital boundaries, and studying how effective these points are for use as landmarks in shape models.

The goal of automatic annotation of shapes in terms of landmarks is to capture the real variability that exists among shapes of the same family in a training set via a model created from the landmarks. Most previous strategies for landmark selection, presented in Section 2.2, lack a robust method that preserves the curvature features of the shapes while achieving the correspondence of landmarks. The c-scale shape descriptor allows the selection of high curvature (valleys in $A_f(b)$) points and/or inflection points (peaks in $A_f(b)$) as mathematical landmarks with different levels of detail and in digital boundaries. Landmarks are detected at different scales to vary the level of detail depending on the application.

Curvature has been commonly used in the past for the location of landmarks, as it appears to be the natural and intuitive approach to characterize a family of shapes [Attneave, 1954]. However, in practice, dominant points do not necessarily lie in the same homologous regions for different shapes of the training set, and it has been demonstrated that global character-

istics of the shapes are more reliable for shape representation [Walker et al., 2002]. Training shapes often present unique features even when representing the same family of shapes. Therefore, we cannot assure that all shapes of the training set will always have the same number of dominant points according to curvature (or other criteria), and be located in corresponding positions among shapes. We believe that the mean shape of a training set contains very important shape information resulting from averaging all shapes. A direct relationship must exist between the mean shape and the shapes in the training set in the sense that most prominent regions of the mean shape appear as a conjunction of all the other shapes. Therefore, finding landmarks on the mean shape will retain the global shape information contained in the training set, instead of searching for specific local characteristics that may appear in only some of the examples of the training set. Then, these landmarks can be propagated to all the instances of the training set to establish landmark correspondence among shapes. In this chapter, we propose several techniques for landmark selection and landmark propagation for establishing landmark correspondence between the mean shape and the shapes in the training set. By combining these techniques of landmark selection and landmark propagation, we can establish the required correspondence to create the shape model.

In this chapter, we first introduce the method for model building in Section 5.1. In Section 5.2, we explain how to align the training shapes by using affine registration. In Section 5.3, we describe how to generate the mean shape of the training set without requiring landmarks. The strategies for the selection of landmarks on the mean shape are presented in Section 5.4. Several strategies for landmark propagation from the mean shape to all the shapes of the training set are presented in Section 5.5. The overall algorithm of the method is given in Section 5.6. Finally, in Section 5.7, we explore the method further by giving some variants of the previous concepts to create the point distribution model (PDM).

5.1 Method of Model Building

In this section we present the overall method (Fig. 5.1) of building the PDM.

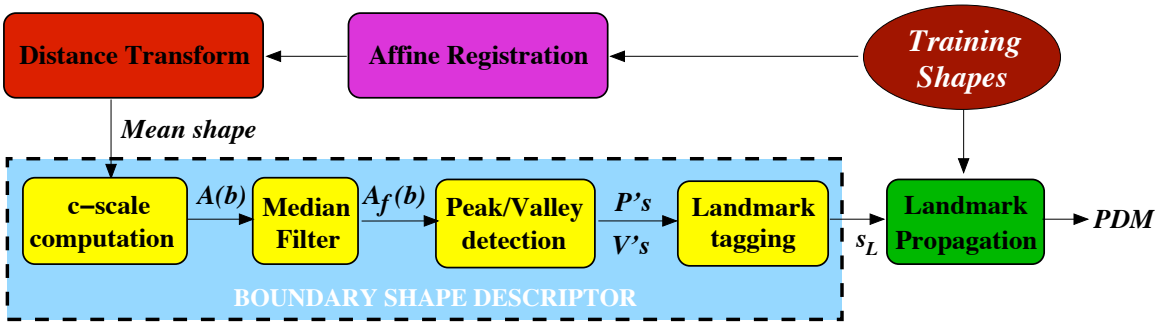


Figure 5.1: The method of model building based on c-scale.

Given a training set of M segmented images, $\{\mathscr{I}_j : j = 1, ..., M\}$, we first align the images by using affine registration. We denote the aligned images in the training set by $\{\mathcal{I}_j : j = 1, ..., M\}$ and extract the boundaries $\{\mathcal{B}_j : j = 1, ..., M\}$ of the aligned segmented structures.

Because we want to capture shape dominant points characterizing the training set and not particular dominant points corresponding to a particular shape in the training set, we locate the landmarks directly on the mean shape boundary, called $\overline{\mathcal{B}}$. The mean shape image $\overline{\mathcal{I}}$ is calculated by using a Euclidean Signed Distance Transform (ESDT) operation on the training set [Souza and Udupa, 2005]. The ESDT is based on the Euclidean Distance Transform (EDT) by [Breu et al., 1995]. Subsequently, we extract the boundary of the mean shape and find the set s_L of landmarks on $\overline{\mathcal{B}}$ by using the c-scale shape descriptor presented in Chapter 3. Then, to generate a PDM, we need to propagate these landmarks to all the shapes of the training set. That way we assure that all the shapes have the same number n of landmarks and in corresponding locations. The approach is summarized in Fig. 5.1.

In the following sections, we explain in more detail some of the essential steps of the method, in particular, shape alignment, mean shape generation, different strategies for land-

mark selection using c-scale, and several approaches for landmark propagation.

5.2 Shape Alignment

According to Kendall [Kendall, 1977], shape can be defined as the geometrical information that remains once we eliminate the location, scale, and rotation components. Therefore, in order to retain only the geometrical shape information, we need to remove from each shape in the training set, the differences due to location, scale, and rotation [Dryden and Mardia, 1998].

We have a set of binary images $\{\mathscr{I}_j : j = 1, ..., M\}$ containing objects of the same family that we need to align in order to retain only the useful geometrical information necessary to build a model. Image registration aims at finding the transformation $\mathcal{T}$ that aligns one image $\mathcal{I}_S$, called source, to another image $\mathcal{I}_T$, called target, to establish a correspondence between them:

$$\mathcal{I}_T = \mathcal{T}(\mathcal{I}_S). \tag{5.1}$$

Affine registration can eliminate the differences due to location, rotation, and scale while preserving the straightness of lines and parallelism, and can be expressed as

$$\mathbf{i}_T = A \cdot \mathbf{i}_S + \ \mathbf{t}, \tag{5.2}$$

where $\mathbf{i}_T$ and $\mathbf{i}_S$ are elements, respectively, from the target image $\mathcal{I}_T$ and the source image $\mathcal{I}_S$, A is a 2×2 matrix containing the transformations of rotation, scaling, and sheering, and $\mathbf{t}$ is a 2×1 vector representing the translations in x and y directions. If we expand this equation further, we have

$$\mathbf{i}_T = \begin{bmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{bmatrix} \mathbf{i}_S + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{5.3}$$

where s includes both the sheering and scaling parameters, θ represents the rotation angle of the transformation, and t_x and t_y are the translations in x and y directions.

To obtain the registered image, the transformation $\mathcal{T}$ needs to be iteratively optimized under a certain similarity metric, which is calculated from all pixels for that particular transformation. There are two classes of similarity metrics available, one based on some functional intensity relationship between images (identity, linear, non-linear), and another assuming a certain statistical relationship between source and target (joint entropy, mutual information, normalized mutual information). The choice of similarity metric may depend on the application, being minimized or maximized, depending on the strategy used, to determine the correct transformation between target and source images. Once the optimal transformation is obtained, some interpolation needs to be done on the source image, after transformation, to get the final registered image.

In our case, we select a target image from the training set. Then, we register to it all the other images of the training set one by one, achieving a pairwise alignment of all shapes with respect to the target. We used an affine registration approach as described in [Periaswamy and Farid, 2003] [Periaswamy, 2003]. This consists of an intensity-based approach, multi-scale (capturing global and local scale transformations), globally smooth and locally affine, and that minimizes the similarity metric, which is the quadratic error function of the intensity values. The Matlab source code used can be found at: *http://www.cs.dartmouth.edu/farid/research/registration.html*.

5.3 Mean Shape Extraction

Our aim is to generate a mean shape from the training shapes without requiring the use of landmarks as in [Souza and Udupa, 2005]. We applied an ESDT, based on the EDT [Breu et al., 1995], to all the aligned binary images forming the training set. As per convention, we define the ESDT as to have negative distance values inside the boundary of each training shape and positive distance values outside the shape boundary. Then, the mean distance map is computed and the mean shape image extracted by thresholding the mean distance map at 0. In the following, we describe the process of mean shape generation using

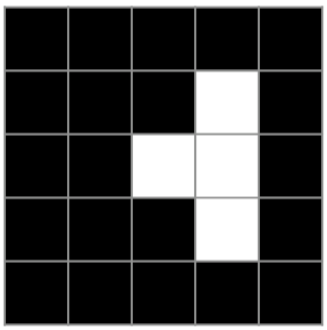
distance transform.

5.3.1 Euclidean Distance Transform

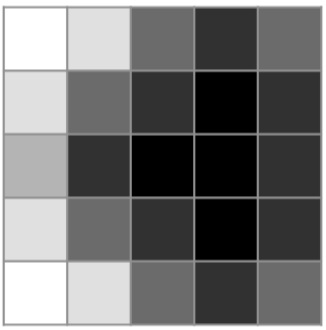
The distance transform aims at finding, for each 0-valued pixel of a binary image, the distance value between that pixel and the nearest 1-valued pixel in the image [Sonka et al., 1999] [Klette and Rosenfeld, 2004]. The Euclidean distance $\mathcal{D}_E$ between two pixels P_1 and P_2 of a binary image, with respective coordinates (x_1, y_1) and (x_2, y_2), is defined as

$$\mathcal{D}_E(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{5.4}$$

Using Equation 5.4, we can create a *distance map* of a binary image, by representing at each pixel, the minimum value of distance between that pixel and the object boundary. That representation generates an image where the object of interest has zero values, the pixels close to the object boundary have small values, and these values increase as we move away from the object boundary. An example of distance map and its associated values can be seen in Fig. 5.2(b)-(c) for the binary image in Fig. 5.2(a).



(a) Binary image.



(b) Distance map.

$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$
$\sqrt{5}$	$\sqrt{2}$	1	0	1
2	1	0	0	1
$\sqrt{5}$	$\sqrt{2}$	1	0	1
$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$

(c) Distance map values.

Figure 5.2: (a) A binary image with its associated (b) distance map and (c) distance values.

[Breu et al., 1995] proposed the Euclidean distance transform used in our study, which is an exact Euclidean distance transform applied to 2D images.

5.3.2 Euclidean Signed Distance Transform

We want to characterize all pixels in a binary image in such a way as to have negative distance values within the object of interest, and positive outside. To achieve this, we used the EDT described in Subsection 5.3.1. The way we create the ESDT, based on the EDT, is shown in Fig. 5.3. First, for a binary image $\mathcal{I}$, we obtain its complement $\mathcal{I}^C$, where the background becomes the object of interest with pixel values of 1, and the remaining pixels having values of 0. We then apply the EDT, defined as a function EDT, to both images $\mathcal{I}$ and $\mathcal{I}^C$, and subtract their values:

$$\mathcal{SDM}(\mathcal{I}) = EDT(\mathcal{I}) - EDT(\mathcal{I}^C), \tag{5.5}$$

where $\mathcal{SDM}$ is the signed distance map of $\mathcal{I}$. The $\mathcal{SDM}$ has negative values inside the object of interest and positive values outside. The values decrease within the object as we move away from the boundary, and they increase outside the boundary in a similar way but with positive values. This gives us a continuous representation of distances along the image and allows us to distinguish interior and exterior of an object, which will be very useful in a number of applications.

5.3.3 Mean Shape Generation and Algorithm

We compute the mean shape of a training set of aligned binary images by applying to each image in the set $\{\mathcal{I}_j : j = 1, ..., M\}$ the ESDT described in Subsection 5.3.2. Once we have the $\mathcal{SDM}$ for each of the images in the training set, we calculate the mean Euclidean signed distance map $\overline{\mathcal{SDM}}$, for each pixel P, such as

$$\overline{\mathcal{SDM}}(P) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{SDM}_j(P), \tag{5.6}$$

where M is the number of binary images in the training set.

Subsequently, we threshold this image to retain all the values smaller than 0. That way we obtain the mean shape binary image and extract the mean object boundary necessary for

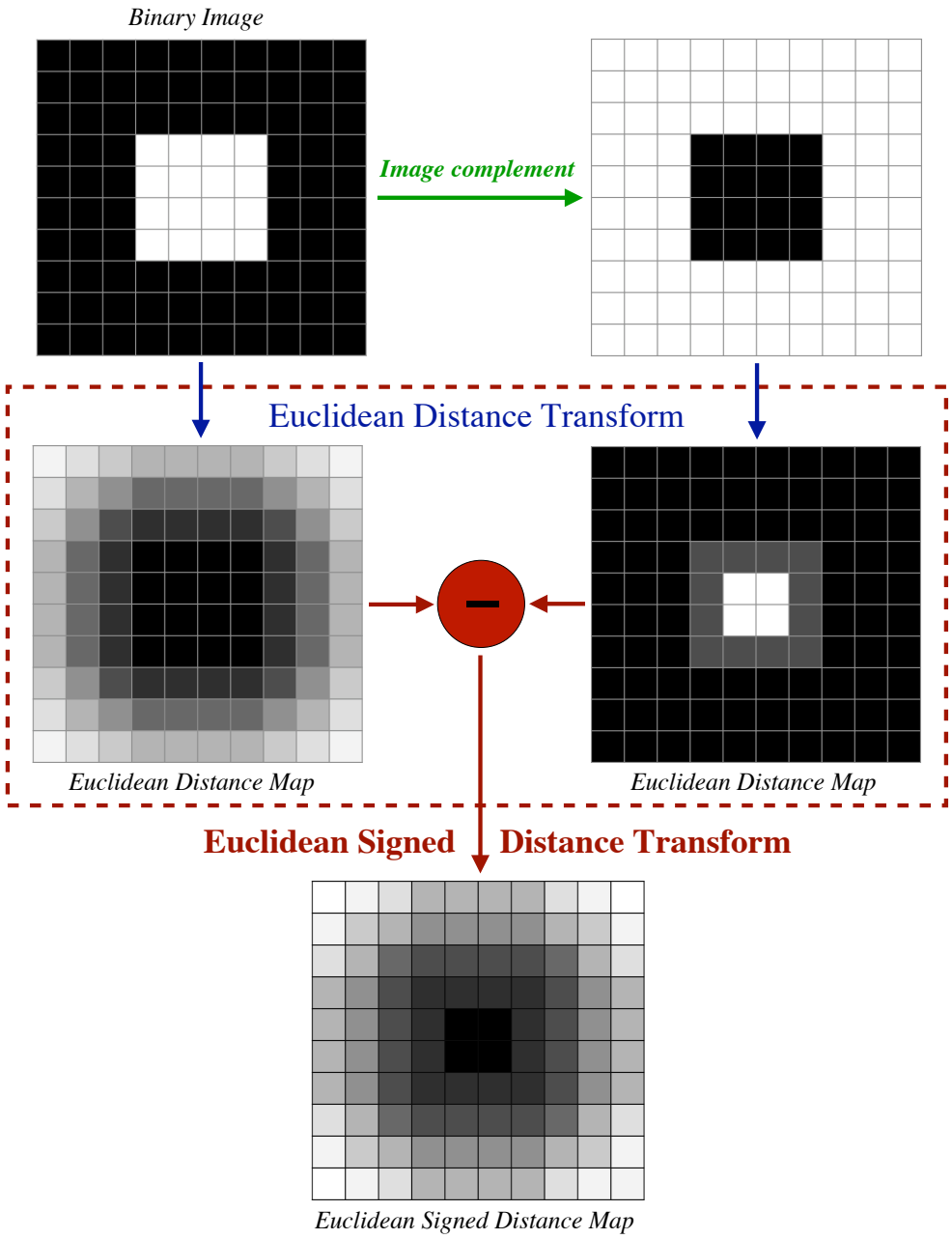


Figure 5.3: Methodology to obtain the ESDT of a binary image using the EDT. Distance map colors vary progressively from black to white. Black represents the lowest distance value and white the highest distance value appearing in the map.

the selection of landmarks. The mean shape boundary corresponds to 0 values in the $\overline{\mathcal{SDM}}$.

The algorithm to obtain the mean shape boundary $\overline{\mathcal{B}}$ from a training set of aligned binary images is as follows:

Algorithm 5.1: **MSE** - Mean Shape Extraction

Input: Binary images $\mathcal{I}_j : j = 1, ..., M$

Output: Mean shape boundary $\overline{\mathcal{B}}$

for $j = 1, ..., M$ **do**

1. Obtain the image complement $\mathcal{I}_j^C$.
2. Obtain $EDT(\mathcal{I}_j)$ and $EDT(\mathcal{I}_j^C)$.
3. Obtain $\mathcal{SDM}_j = EDT(\mathcal{I}_j) - EDT(\mathcal{I}_j^C)$.

end

4. Calculate $\overline{\mathcal{SDM}}$ by using Equation 5.6.
5. Threshold $\overline{\mathcal{SDM}}$ at 0 to create the mean binary image $\overline{\mathcal{I}}$.
6. Extract the mean shape boundary $\overline{\mathcal{B}}$ from $\overline{\mathcal{I}}$.

5.4 Hierarchical and Non-Hierarchical Landmark Selection

In this section, we describe several ways of selecting the set s_L of landmarks on $\overline{\mathcal{B}}$. Landmarks can be selected in a *hierarchical* or *non-hierarchical* way.

In hierarchical selection, the steps are as follows:

1. **Fix the parameters** t, w, m, se_v, se_p, θ_v and θ_p for a certain application to get the maximum number of landmarks (peaks and/or valleys) needed to create a model.
2. **Arrange peaks in descending order** by using their corresponding $A_f(b)$ values.
3. **Arrange valleys in ascending order** by using their corresponding $A_f(b)$ values.
4. Select as many **prominent valleys and/or peaks** as the number of landmarks desired.

In this manner, we can select the peaks and valleys hierarchically from the most prominent to the least. This allows us to select precisely the desired number n of landmarks for an application by just considering only the n most prominent peaks and/or valleys. Higher peaks in $A_f(b)$ correspond to longer straight regions, whereas lower valleys correspond to regions of higher curvature. This strategy is especially useful if the process of finding landmarks needs to be repeated many times, for example, if the number of landmarks needs to be varied repeatedly for a particular application or evaluation study.

As opposed to the above hierarchical approach, we can also select n landmarks by adjusting the parameters (particularly, t, se_v, se_p, θ_v, and θ_p) for a particular application, to obtain the desired number of landmarks on the mean shape. We call this method non-hierarchical.

Once we have detected the peaks and valleys, we locate the landmarks on the boundary of the shape by considering only valleys, only peaks, or both. This gives us overall six possibilities for selecting landmarks.

5.5 Landmark Propagation

The last step of the method consists of propagating the landmarks identified on the mean shape to all boundaries of the training set. In the following, we introduce two strategies for landmark propagation: *closest point propagation* and *parametric propagation*.

5.5.1 Closest Point Propagation

In the *closest point propagation* approach, for each landmark P in the mean shape, the point identified on a training shape is that point on its boundary, which is closest to P (Fig.5.4).

Thus, once we have the landmarks tagged on the mean shape (Fig.5.5(a),(b)), we can propagate all the landmarks by finding, for each landmark in the mean shape, the closest point on the training shape boundary as shown in Fig.5.5(c). This process automatically establishes landmark correspondence among shapes in the training set.

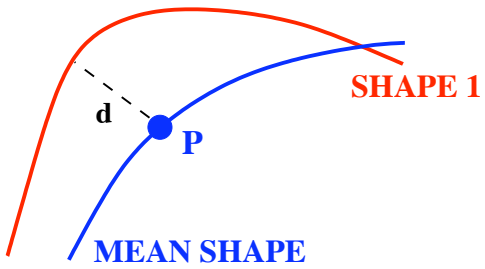


Figure 5.4: Closest Point Propagation.

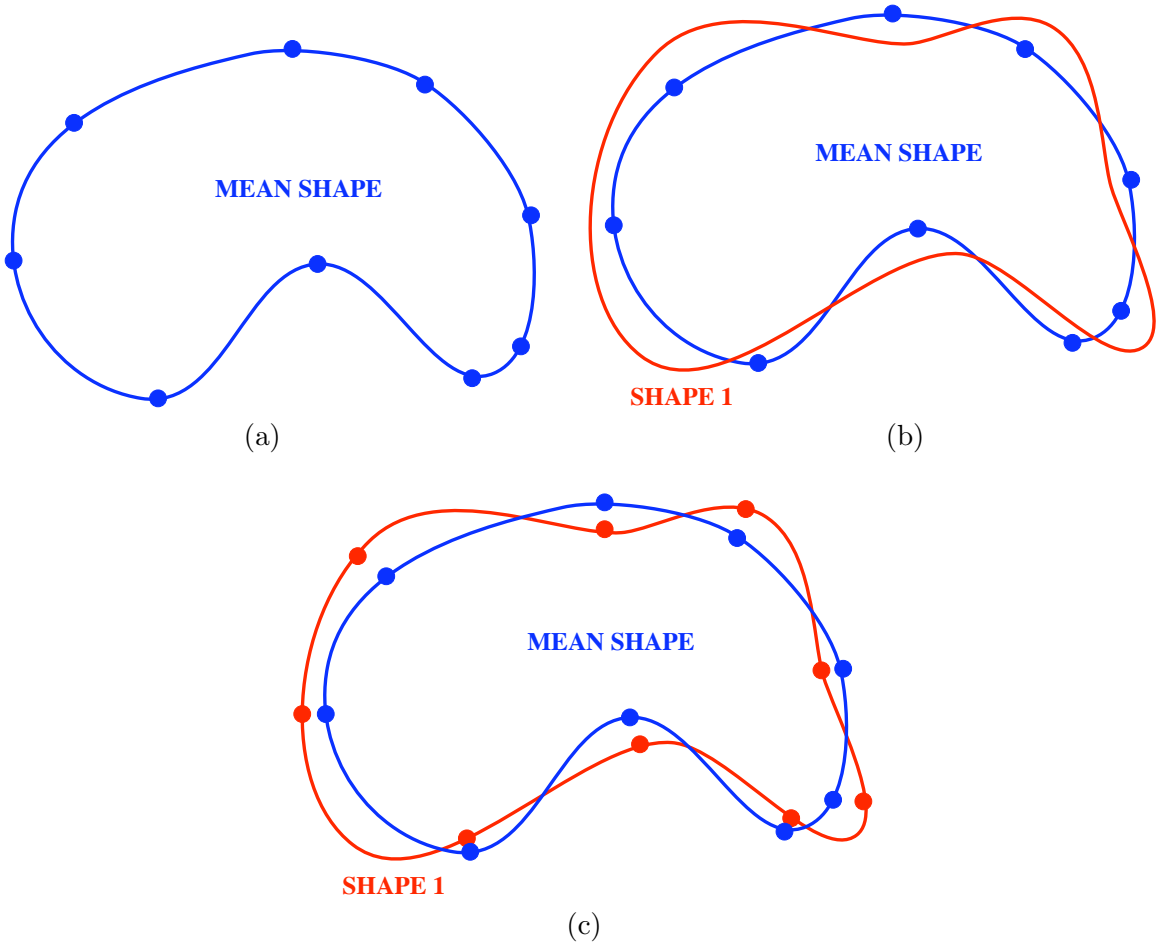
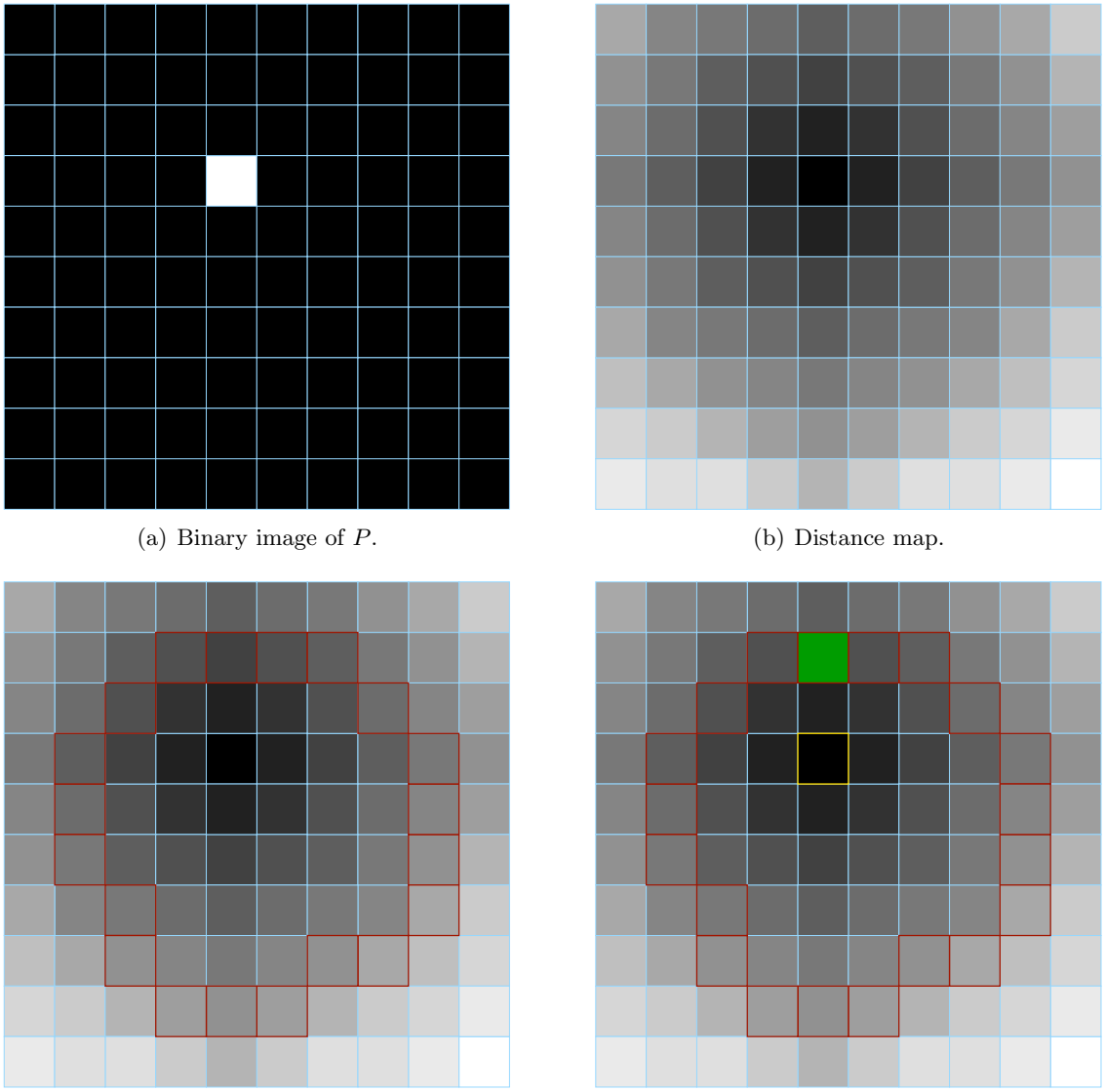


Figure 5.5: (a) Mean shape with landmarks tagged. (b) Contour of a training shape overlayed on the mean shape. (c) Closest point propagation of landmarks.

Implementation Details and Algorithm

The identification of landmarks in the training shapes, from a knowledge of the landmarks in the mean shape, is accomplished via EDT [Breu et al., 1995]. For each landmark P selected



(a) Binary image of P.

(b) Distance map.

(c) Overlayed training shape.

(d) Selection of closest point.

Figure 5.6: Closest Point propagation of a landmark P to a training shape by using EDT. The training shape bels are outlined in red and the pixel P in yellow. The resulting propagated landmark is represented in green in (d).

on $\overline{\mathcal{B}}$, we create a binary image, of the same size as the training set images, where the pixel corresponding to P is assigned a value 1 and the rest of the pixels are assigned the value 0, as shown in Fig. 5.6(a). Then, by applying the EDT to the binary image, we obtain the associated distance map as represented in Fig. 5.6(b).

We find the location of the bels, for each training shape, in the distance map as shown in Fig. 5.6(c) by the red outlined pixels. We generate a vector V for each shape and for each landmark, which contains the associated distance values at each bel taken from the distance map. By finding the minimum value of the vector we can determine the closest point from P to the shape and propagate the landmark. In the example in Fig. 5.6(d), the landmark P is outlined in yellow and the pixel associated with the propagated landmark is represented in green. We repeat the process for each landmark $P_1, ..., P_n$ on the mean shape and obtain the propagated landmarks $p_1^j, ..., p_n^j$ for each training shape j and repeat the process for $j = 1, ..., M$.

The algorithm used in closest point propagation is given below.

Algorithm 5.2: **CPP** - Closest Point Propagation

Input: Mean shape landmarks $P_i : i = 1, ..., n$.

Shape boundaries $\mathcal{B}_j : j = 1, ..., M$.

Output: Landmarks $p_i^j : i = 1, ..., n; j = 1, ..., M$.

for $i = 1, ..., n$ **do**

1. Generate a binary image from P_i.
2. Apply the EDT to that binary image and obtain the distance map.

for $j = 1, ..., M$ **do**

3. Store the distance values for all the elements of $\mathcal{B}_j$ in V_j^i.
4. Output $p_i^j = P$, where P is the pixel with minimum value in V_j^i.

end

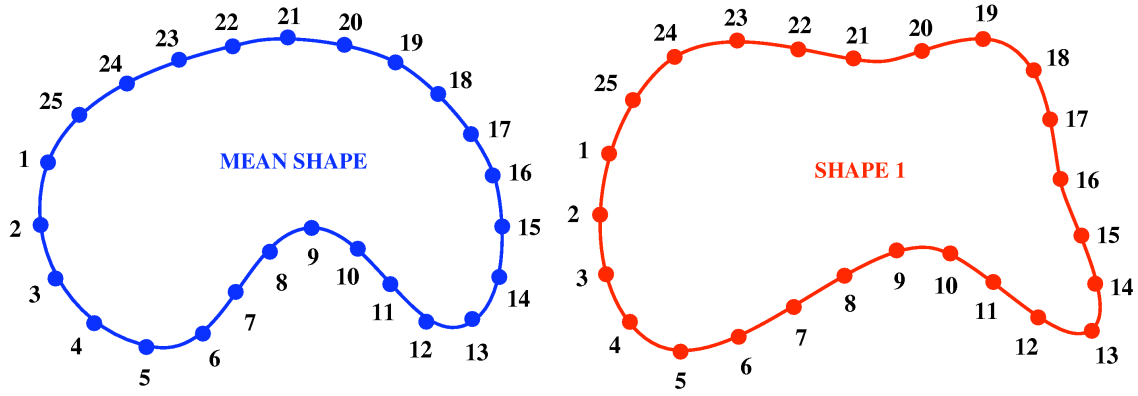
end

5.5.2 Parametric Propagation

The *parametric propagation* strategy consists of the following steps:

1. Reparameterize the boundary shapes of the training set guided by the shape with

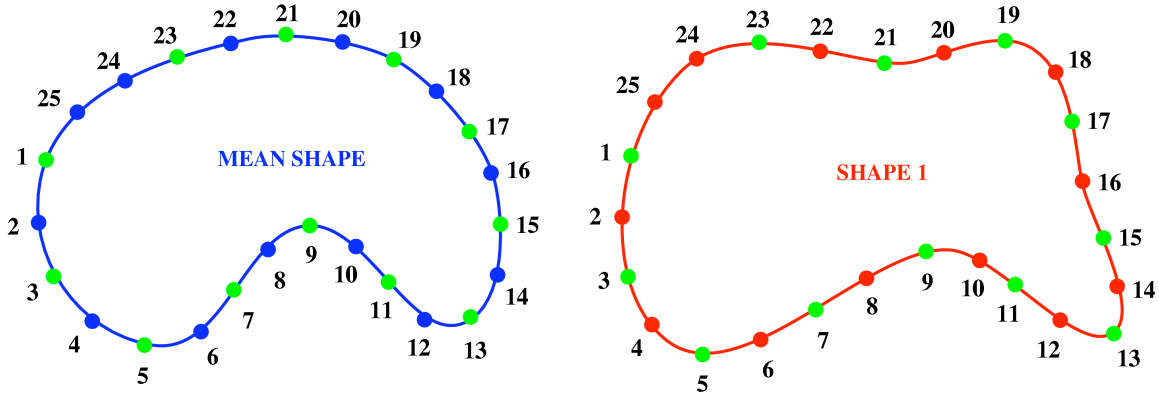
the minimum number g of bels. Each training shape boundary and the mean shape boundary are resampled so as to have exactly g equally spaced points. The initial points on all shape boundaries need to be defined manually at corresponding positions. This stage is illustrated in Fig. 5.7.



(a) Mean shape. (b) Training shape.

Figure 5.7: Parameterized shapes: all shapes have the same number of bels equally spaced along the contours. Blue and red points on the contours denote bels.

2. Find the appropriate landmarks on the mean shape by using the c-scale approach as shown in Fig. 5.8(a).



(a) (b)

Figure 5.8: (a) Landmarks found on the mean shape (represented by green points). (b) Landmarks propagated to the training shape from the mean shape by using the parametric positions.

3. Propagate the landmarks from the mean shape to the shape boundaries of the training set by placing them at their corresponding parametric positions. This is illustrated in Fig. 5.8(b) for the mean shape with landmarks in Fig. 5.8(a). Landmarks are represented by green points.

The algorithm used for parametric propagation is presented below.

Algorithm 5.3: **PP** - Parametric Propagation

Input: Shape boundaries $\mathcal{B}_j : j = 1, ..., M$; Mean Shape $\overline{\mathcal{B}}$.

Output: Landmarks $p_i^j : i = 1, ..., n; j = 1, ..., M$.

1. Find the minimum number of bels g over all $\mathcal{B}_j : j = 1, ..., M$, and $\overline{\mathcal{B}}$.
2. Define a starting and corresponding point on all $\mathcal{B}_j : j = 1, ..., M$, and $\overline{\mathcal{B}}$.
3. Reparameterize all $\mathcal{B}_j : j = 1, ..., M$ and $\overline{\mathcal{B}}$, so each contour has g equally spaced bels.
4. Find landmarks on the parameterized mean shape by using the c-scale method.
5. **for** *j=1,...,M* **do**
   Propagate the landmarks from the parameterized mean shape to the parameterized boundaries of the training set, at their corresponding parametric positions.
   end
6. Output $p_i^j : i = 1, ..., n; j = 1, ..., M$.

5.6 Overall Algorithm

To summarize the whole process described in this chapter, we present the overall algorithm for the c-scale based landmark tagging process, using the algorithms presented in the previous sections.

Algorithm 5.4: **CSB** - Curvature Scale-Based Landmark Tagging

Input: Binary images $\mathscr{I}_j : j = 1, ..., M$; c-scale parameters: t, w, m, se_v, se_p, θ_v, and θ_p.

Output: Landmarks $p_1^j, ..., p_n^j, j = 1, ..., M$.

1. For $j = 1, ..., M$, align all binary images $\mathscr{I}_j$ to the target binary image $\mathscr{I}_T \in \{\mathscr{I}_j : j = 1, ..., M\}$ by using affine registration to yield $\{\mathcal{I}_j : j = 1, ..., M\}$.
2. Calculate the mean shape boundary $\overline{\mathcal{B}}$ by using Algorithm 5.1.
3. Extract the boundary of the training shapes $\{\mathcal{B}_j : j = 1, ..., M\}$ from $\{\mathcal{I}_j : j = 1, ..., M\}$.
4. Find the landmarks $P_i, i = 1, ..., n$ on $\overline{\mathcal{B}}$ by using the boundary shape descriptor based on c-scale.
5. Propagate the landmarks $P_i, i = 1, ..., n$ from $\overline{\mathcal{B}}$ to all training shapes $\{\mathcal{B}_j : j = 1, ..., M\}$ by using Algorithm 5.2 or Algorithm 5.3.
6. Return $p_1^j, ..., p_n^j, j = 1, ..., M$.

5.7 Variants of the Method

We note that, from the three types of landmarks (peaks only, valleys only, peaks and valleys), identified hierarchically or non-hierarchically, and from the two approaches of propagation, we have described 12 possible methods of tagging landmarks altogether as represented in Table 5.1.

Table 5.1: Landmark Tagging Methods

Peaks	**Valleys**	**Peaks and Valleys**
NH,CP	NH,CP	NH,CP
NH,P	NH,P	NH,P
H,CP	H,CP	H,CP
H,P	H,P	H,P

NH: Non-Hierarchical; H: Hierarchical; CP: Closest Point; P: Parametric.

With all landmarks located in all shapes of the training set, in equal number n and in corresponding positions, we can now create the PDM.

The use of both peaks and valleys for landmark selection can also be specified in different ways. For example, we could use all peaks and valleys as landmarks, half-half, or any other combination considering complementary percentages. The selection of the type of landmark to use, as well as the combinations, depend on the application.

5.8 Conclusion

In this chapter, we have presented a general framework to find landmarks using salient points. Any other landmark propagation method or shape descriptor could be used as well instead of the ones presented here. In this work, we have focused on using the c-scale shape descriptor because it has many benefits, from being multi-scale, directly applicable to digital boundaries, and presenting complementary information about the characteristics of each landmark (concave, convex, flat), which could be very useful in many applications. Different variants of the method will be evaluated in Chapter 8.

6

Building 2D Shape Models Via Shape Variance Equalization

Although the *c*-scale concept brought on several new ideas which led to considerable improvements in salient point detection from the state-of-the-art, since the same dominant points may not occur in all training shapes, it poses a hurdle for the *c*-scale based landmark tagging strategies. In an effort to overcome this hurdle, in this chapter we propose a salient-point-free landmark tagging approach which selects landmarks guided entirely by the shape variance that exists in the training set.

It is well known that model-based segmentation approaches, such as those employing ASMs [Cootes et al., 1995], require a complete training set of shapes characterizing as much as possible the variability existing for a particular object in a population. If the natural variability of the object is included in the model to use for segmentation, then, the model will adapt better when segmenting objects of the same family. However, if the strategy used to select landmarks on shapes does not take into account the real variability existing in the training set, the resulting model will not incorporate all the information necessary to fully characterize that object. For example, in Fig. 6.1, we have represented three different 2D samples of the talus bone of the foot belonging to three different individuals. As we can notice, the top part of the shape is very similar among all three shapes, whereas the bottom part has a high degree of variance among shapes. Therefore, to adequately characterize this training set, more landmarks will have to be selected in the bottom part of the shapes than

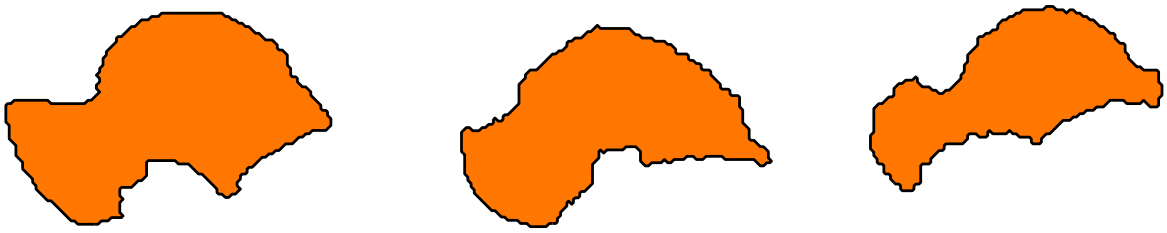


Figure 6.1: Shapes of the talus bone of the foot from different individuals.

in the top part for the model to capture the full variability existing in the training set.

The main premise of our approach in this chapter is that regions with high variance among shapes will need more landmarks to be described consistently, whereas regions with low variance do not need to be described with many landmarks. Therefore, our attitude will be to distribute landmarks around each shape in such a manner as to best capture the shape variation existing in the training set. This is achieved by placing more landmarks in regions where the variance is high and fewer landmarks where it is low. This strategy itself takes care of the correspondence issue, and at the same time, deploys landmarks very frugally and optimally considering shape variations. The desired landmarks are positioned around each shape boundary so as to equally distribute the total variance existing in the training set in a global-to-local manner. The landmarks may not correspond to dominant points on the shape as in previous strategies, but they will give a good representation of the variability existing in the training set. This is a consideration that has been missing in all landmark tagging methods proposed up to now. The variance equalization method is simple to implement, computationally fast, and does not need a reference shape to establish landmark correspondence as needed in, for example, methods based on optimizing Minimum Description Length [Davies et al., 2002] [Thodberg, 2003a].

In this chapter, we present a method for automatic landmark tagging that uses a strategy of equalization of the total variance existing in the training set, automatically establishing

correspondence among shapes at the same time. We give an overview of the method in Section 6.1. In Section 6.2, we introduce the parameterization concept. In Section 6.3, we explain the overall concept of variance equalization. The selection of landmarks via variance equalization is explained in Section 6.4. The overall algorithm is presented in Section 6.5.

6.1 Overview of the Method

The method of variance equalization consists of four main parts as shown in Fig. 6.2.

Figure 6.2: The method of Variance Equalization.

Given a training set of M segmented images, we align them by using affine registration and extract the boundaries $\{\mathcal{B}_j; j = 1, ..., M\}$ of the represented structures. Then, we parameterize the boundaries so as to have the same number of bels, equally spaced, for each shape. Subsequently, we calculate the total variance among shapes for each new bel position in the parameterized boundaries, and locate landmarks on the shapes by equalizing the variance along the boundaries. More details about the parameterization and variance equalization steps are given in the following sections.

6.2 Parameterization

After identifying the corresponding starting point in all registered shape boundaries of the training set, we *parameterize* the boundaries so that they have the same number N of points equally spaced along each boundary. We take N to be the minimum number of boundary points among all boundaries. We do so because we wish to retain as much information in the original boundaries as possible. We thus obtain a parametric description of the boundaries $\{\mathcal{B}_j(i); i = 1, ..., N; j = 1, ..., M\}$. In Fig. 6.3 we show an example of parameterization of two

shapes for $N = 27$. The initial point has been manually placed at corresponding locations.

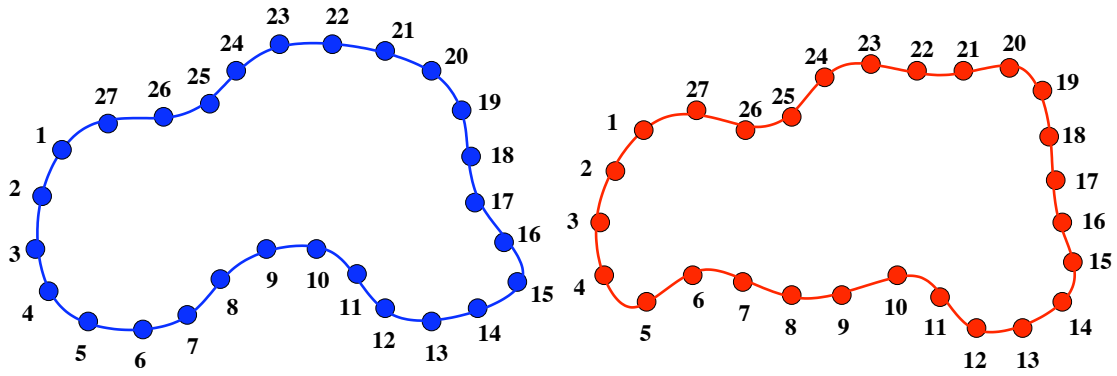


Figure 6.3: Parameterization of two shapes. Both shapes have corresponding starting points, and the same number of bels, equally spaced, on each boundary.

6.3 Variance Equalization

From the M parameterized boundaries $\{\mathcal{B}_j(i); i = 1, ..., N; j = 1, ..., M\}$, we can determine the variance in the location of any i^{th} point, $i = 1, ..., N$, and express it as a function $Var(i)$ as shown in Fig. 6.4. The total variance at each i^{th} point over all M shapes can be estimated by Principal Component Analysis of the (x_i^j, y_i^j) coordinates of this point over all M parameterized shapes. The sum of the $2M$ eigenvalues $\lambda_i^k, k = 1, ..., 2M$, of the covariance matrix of the x_i^j and y_i^j coordinates of the i^{th} point over all M shapes yields $Var(i)$.

$$Var(i) = \sum_{k=1}^{2M} \lambda_i^k, i = 1, ..., N. \tag{6.1}$$

Our goal is to place landmarks by making use of this information in $Var(i)$. We wish to select more landmarks in regions where $Var(i)$ is higher, and less where it is small. That is, we will use the total variance distribution to guide the selection of new points so that the variance is more or less equally distributed with point density on the shape boundaries.

Once the number n of landmarks we want in the model is specified, we distribute the

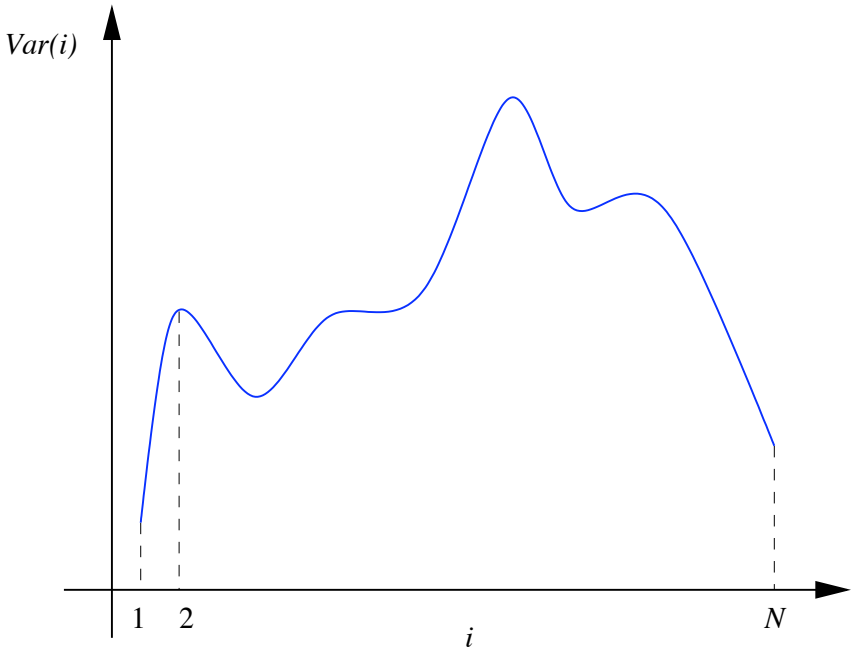


Figure 6.4: Variance curve.

landmarks around each boundary in such a way that the area under the variance curve between each successive pair of landmarks is roughly the same. The selection of landmarks will be done separately for each boundary and each will get its point arrangement depending on its shape and the boundary's relationship to the variance graph. The details of this process are presented below.

6.4 Landmark Selection Via Variance Equalization

Let x_i, $i = 1, ..., N$, denote the i^{th} point generically in any parameterized $\mathcal{B}_j$, let $Var(x_i)$ be the total variance at x_i, and let del_j be the spacing between points in $\mathcal{B}_j$ as shown in Fig. 6.5. Each $\mathcal{B}_j$ may have a different del_j, which needs to be taken into account.

Given $Var(x_i)$, n, and del_j, we wish to find the points $p_1, ..., p_n$ on $\mathcal{B}_j$ such that

$$\begin{aligned} &Var(p_1) \cdot |p_2 - p_1| + \ldots + Var(p_n) \cdot |p_1 - p_n| = \\ &Var(x_1)del_j + Var(x_2)del_j + \ldots + Var(x_N)del_j, \end{aligned} \tag{6.2}$$

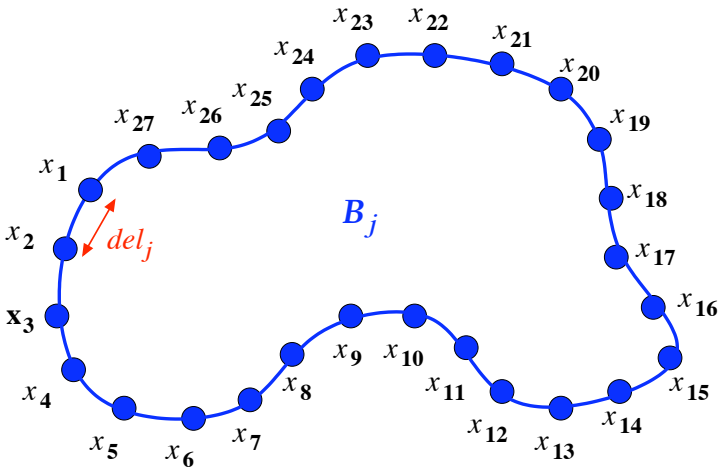


Figure 6.5: $\mathcal{B}_j$ has a spacing del_j between successive points in the contour.

where $|\cdot|$ denotes the distance along the boundary $\mathcal{B}_j$ between successive points. Observe that, $|p_2 - p_1| = (e+1) \cdot del_j$, if there are e points in $\mathcal{B}_j$ between p_1 and p_2 excluding p_1 and p_2. The right side of (6.2) corresponds to the total variance, denoted TA_j, in $\mathcal{B}_j$ over all its points, which is also the area under the $Var(x_i)$ curve, defined as

$$TA_j = Var(x_1)del_j + Var(x_2)del_j + \ldots + Var(x_N)del_j. \tag{6.3}$$

Each boundary $\mathcal{B}_j$ will have an associated TA_j.

After selecting n, we can compute the area per each point p_k, $k = 1, \cdots, n$, denoted A_k^j, for each $\mathcal{B}_j$ as

$$A_k^j = \frac{TA_j}{n}, \tag{6.4}$$

where A_k^j is also the partial area between successive landmarks. The first landmark, p_1^j, will be selected on each contour as the point on $\mathcal{B}_j$ at which $Var(x_i)$ is maximum, and as shown in Fig. 6.6.

From p_1^j, we go forward on $\mathcal{B}_j$ and skip the parameterized points until we reach p_2^j such

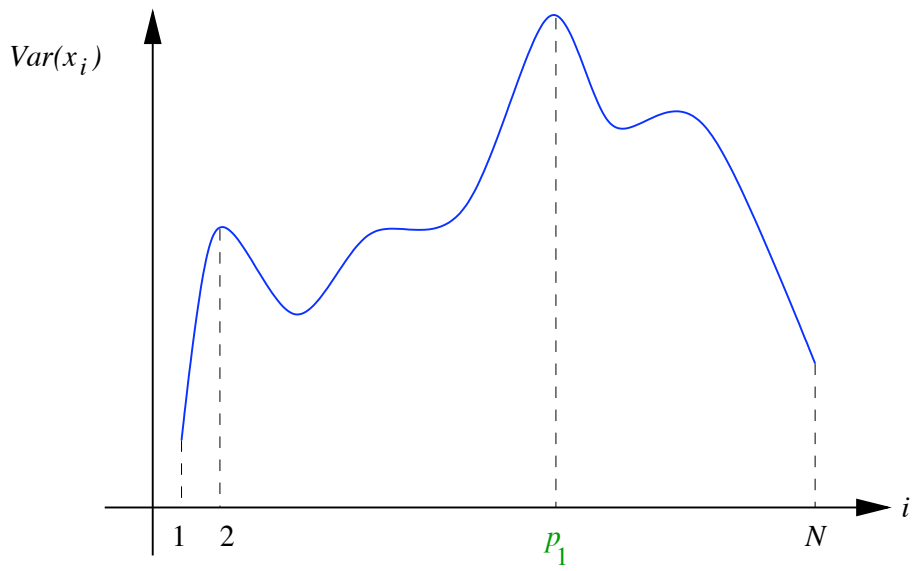


Figure 6.6: Selection of p_1^j on $Var(x_i)$ curve.

that

$$Var(p_1^j) \cdot |p_2^j - p_1^j| \approx A_1^j. \tag{6.5}$$

This process is also illustrated in Fig. 6.7. Once p_2^j is found, we go forward from p_2^j and

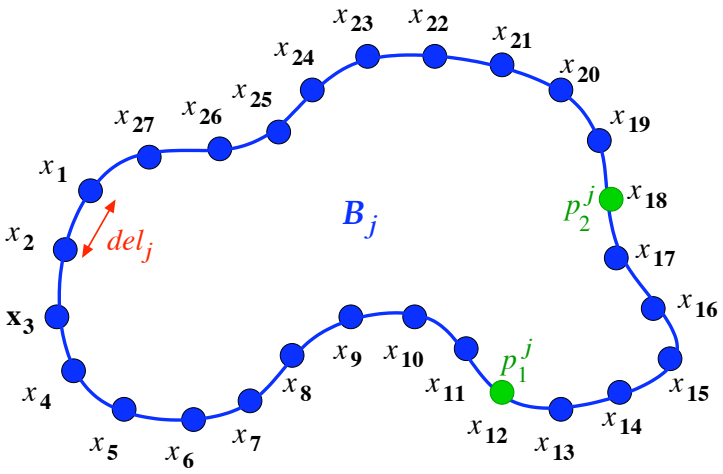


Figure 6.7: Selection of p_2^j on $\mathcal{B}_j$.

repeat this process until we come back to p_1^j, finding all the landmarks $p_1^j, \cdots, p_n^j$ for all $\mathcal{B}_j$. In general, to find a new landmark p_k^j on $\mathcal{B}_j$, from a knowledge of p_{k-1}^j, and with $p_k^j \neq p_1^j$,

we need to satisfy

$$Var(p_{k-1}^j) \cdot |p_k^j - p_{k-1}^j| \approx A_k^j. \tag{6.6}$$

The correspondence between landmarks among shapes is automatically taken care of by distributing the landmarks on each boundary separately but following the graph indicated by the $Var(x_i)$ curve. The overall algorithm for this method is presented below.

6.5 Algorithm VE

The algorithm used to implement the variance equalization method is summarized below.

Algorithm 6.1: **VE** - Variance Equalization Landmark Tagging

Input: $\mathcal{B}_j(i) : j = 1, ..., M; i = 1, ..., N$; n; del_j.

Output: Landmarks $p_1^j, ..., p_n^j, j = 1, ..., M$.

1. Compute $Var(x_i)$, for $i = 1, ..., N$.

for $j = 1, ..., M$, **do**

2. Compute the area A_k^j per point p_k, $k = 1, ..., n$, for $\mathcal{B}_j$ by using (6.4).
3. Select the first landmark on $\mathcal{B}_j$ as the point on $\mathcal{B}_j$ at which $Var(x_i)$ is maximum. Call this point p_1^j.
4. From p_1^j, go forward on $\mathcal{B}_j$ incrementally and skip points until we reach p_1^j back again such that (6.5) is satisfied, thus selecting all n points on $\mathcal{B}_j$.
5. Output $p_1^j, ..., p_n^j$.

end

6.6 Conclusion

In this chapter, we have presented a new theory and method for automatic landmark tagging guided by the strategy of equalization of the variance contained in a training set for selecting landmarks. The main premise here is that this strategy itself takes care of the correspondence

issue and at the same time deploys landmarks very frugally and optimally considering shape variations. The desired landmarks are positioned around each contour in such a manner as to equally distribute the total variance existing in the training set. This concept has been missing in all previous attempts at automatic landmark tagging. The evaluation of this method is presented in Chapter 8.

7

Building 2D Shape Models Via Recursive Boundary Subdivision

The techniques discussed up to this point have been developed for 2D shapes only, and their generalization to three- and higher-dimensional spaces is not straightforward. However, these are very useful techniques since 2D modelling has numerous applications. In order to overcome the generalizability hurdle, in this chapter, we propose a novel global-to-local strategy that directly addresses the generalization of methods to $\mathfrak{n} > 2$ dimensions, while making computations practical, and operating in $\Re^{\mathfrak{n}}$. It attempts to mathematically define and characterize landmarks and ensures the homology among landmarks via recursive boundary subdivision (*RBS*) of all training shapes simultaneously, maintaining correspondence among shapes and their subdivided segments, as per a specified shape similarity criterion, at every recursive step. For the actual selection of landmarks on the sub-segments any of numerous techniques, including those from c-scale and variance equalization, can be used. The 2D version starts from two initial corresponding points determined in all training shapes via a method α, and subsequently by subdividing the shapes into connected boundary segments by a line determined by these points. A shape analysis method β is applied on each segment to determine a landmark on the segment. This point introduces more pairs of points, the lines defined by which are used to further subdivide the boundary segments. This RBS process continues simultaneously on all training shapes, maintaining synchrony of the level of recursion, and thereby keeping correspondence among generated points automatically. The process terminates when no subdividing lines are left to be considered that indicate (as per

method β) that a point can continue to be selected on the associated segment. The method is computationally simple and does not require registration but uses the fundamental attributes of the shapes in the training set. It undertakes the selection of landmarks in a coarse-to-fine manner, consulting all objects in the training set simultaneously in establishing homology and in finding landmarks. In some manner, this approach incorporates the concepts introduced in the c-scale (Chapter 5) and variance equalization (Chapter 6) approaches, in the sense that it characterizes a training set of shapes with as many shape features as are present among shapes of the training set. The whole method operates on discrete curves (or surfaces) to locate landmarks. Connectivity of the curves (surfaces) is required for the algorithm.

In Section 7.1, we describe the concepts of digital objects and boundaries that form the basis of RBS. Section 7.2 describes the principles and algorithm of RBS. In Section 7.3, we present some boundary landmark selection strategies. A novel concept of virtual landmarks is introduced in Section 7.4. The hierarchical nature of the RBS method is discussed in Section 7.5.

7.1 Digital Objects and Boundaries

The geometric and topological properties of digital oriented, connected Jordan surfaces form an essential underpinning of the proposed RBS method. A Jordan surface S is such that:

1. It divides the space Y, in which it is defined, into two sets E and F, which are such that $Y = E \cup F$;

2. Any path from a point P in F to a point Q in E will meet S.

The same concept is applicable to curves. In this section, we first present some information about the concepts of connectedness, orientation, and topological properties of curves and surfaces. See [Udupa, 1994] for more details.

7.1.1 Spatial Elements: Spels

We call the elements of the digital space defined by $\mathbb{Z}^{\mathfrak{n}}$, $\mathfrak{n} \geqslant 1$, spatial elements or *spels*. Spels are pixels for $\mathfrak{n} = 2$, voxels for $\mathfrak{n} = 3$, and hypervoxels for $\mathfrak{n} = 4$, as represented in Fig. 7.1. All images we deal with are digital and are defined in the digital space.

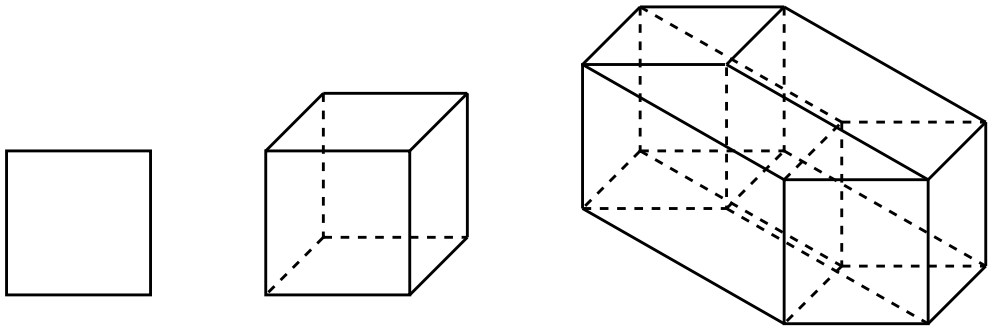


Figure 7.1: Elements of $\mathbb{Z}^{\mathfrak{n}}$: pixels (2D), voxels (3D), and hypervoxels (4D).

7.1.2 Object and Co-object

An *object* in a binary image is a connected component of spels. Many choices are available for defining object connectivity, the choice increasing with $\mathfrak{n}$. For example, for $\mathfrak{n} = 2$, we can define 4- and 8-connectivity. For $\mathfrak{n} = 3$, we can have 6-, 14-, 18-, and 26-connectivity, etc. Similarly, background (co-object) connectivity can also be independently defined. In the same binary image, object and co-object connectivities should be selected complementarily to avoid conflicting situations. In the following, let Υ and Φ denote the object and background connectivity relations, respectively. To illustrate the importance of the connectivity selection, consider the objects and co-objects for different configurations of Υ and Φ in Fig. 7.2. If we define Υ = 4-connectivity for object and Φ = 8-connectivity for co-object, we obtain the result shown in Fig. 7.3. With this configuration, we find two objects, represented in green and in red in Fig. 7.3(a) and one co-object in cyan as shown in Fig. 7.3(b). Besides, if we select Υ = 8-connectivity for object and Φ = 4-connectivity for co-object, we obtain the result shown in Fig. 7.4. In this case, we identify only one object, shown in green in

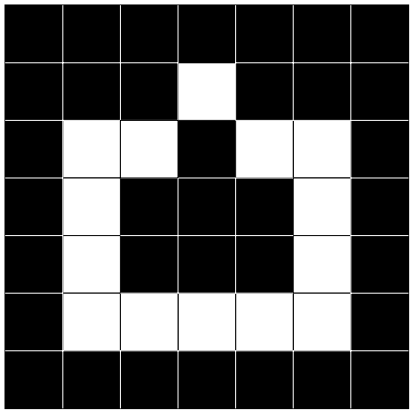
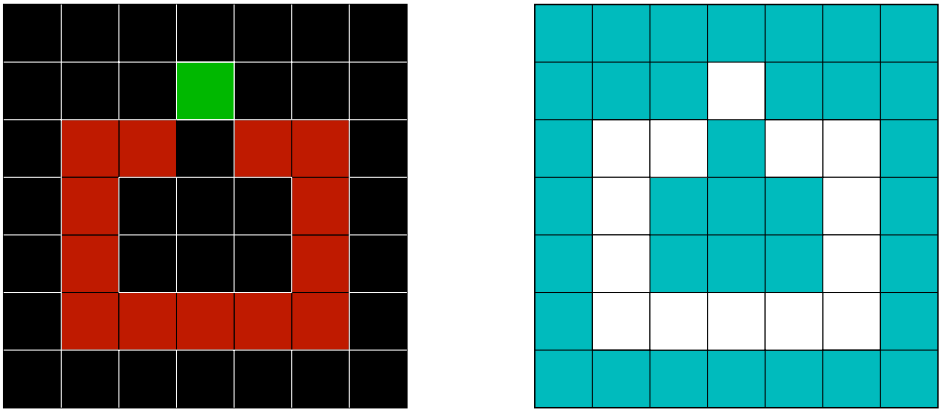
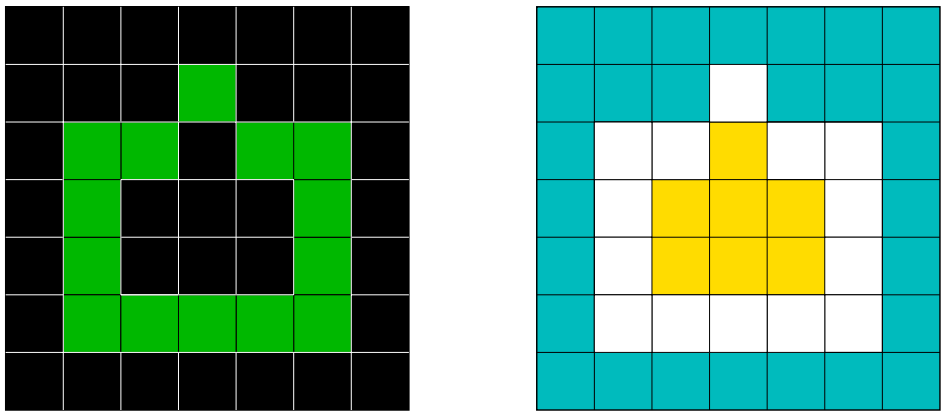


Figure 7.2: A binary image formed by 2D spels or pixels. White spels correspond to pixels with value 1, and black spels are pixels with value 0. The number of objects and co-objects in this figure depends on the connectivity definition.



(a) Objects (b) Co-object

Figure 7.3: Connected components of spels for $\Upsilon = 4$-connectivity and $\Phi = 8$-connectivity. (a) Objects. (b) Co-object.



(a) Object (b) Co-objects

Figure 7.4: Connected components of spels for Υ = 8-connectivity and Φ = 4-connectivity. (a) Objects. (b) Co-object.

Fig. 7.4(a), and two co-objects, represented in yellow and cyan in Fig. 7.4(b).

For $\mathfrak{n} = 3$, for the set of voxels represented in Fig. 7.5, we obtain the whole set of spels as an object (red spels) if we consider Υ = 18-connectivity.

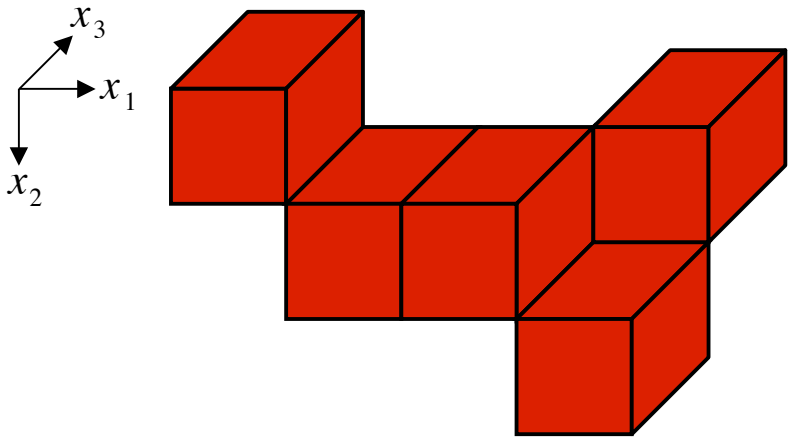


Figure 7.5: Connected components of spels for Υ = 18-connectivity.

If we define Υ = 6-connectivity, then we obtain 4 objects as shown in Fig. 7.6. Each object is represented in a different color.

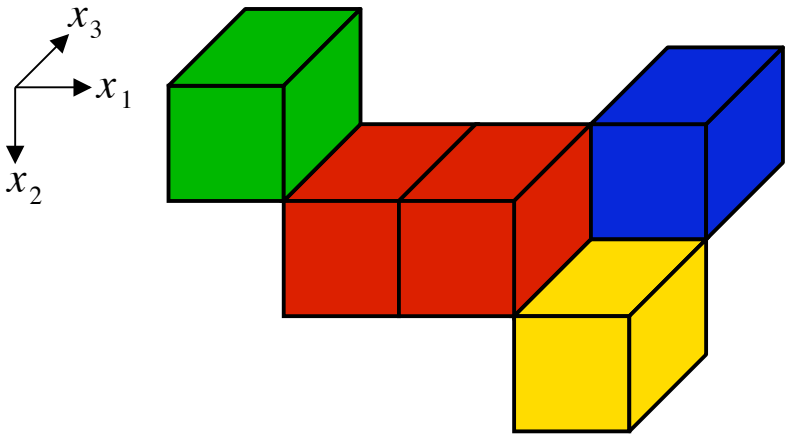


Figure 7.6: Connected components of spels for Υ = 6-connectivity.

The tightest form of adjacency, namely 4-adjacency for $\mathfrak{n} = 2$, 6-adjacency for $\mathfrak{n} = 3$, etc., plays a central role in our definition of oriented surfaces. We shall call this 1-adjacency, because, in such a case, adjacent spels differ in exactly one of their coordinates by 1.

7.1.3 Boundary Elements: Bels

The next important concept we need is the concept of a boundary element or *bel*. A bel of an object O is an ordered pair of spels (o, u) such that $o \in O$ and u is a spel of the background, and o and u are $1-$adjacent. For $\mathfrak{n} = 2$, a bel is an oriented pixel edge, as shown in Fig. 7.7(a). The orientation on each edge is represented by a normal vector to the edge going from o out into u. In Fig. 7.7(a), the red edge has a normal vector indicated in red and called b_1, which points in the direction of the x_1-axis. Similarly, the blue edge is represented by the vector b_{-2}, which points in the direction opposite to the x_2-axis.

For $\mathfrak{n} = 3$, a bel represents an oriented face of a voxel (see Fig. 7.7(b)), the orientation (like a normal vector to the face) going from o out into u. The name used to denote the normal vector depends on its orientation with respect to the axis defined for that object in that direction.

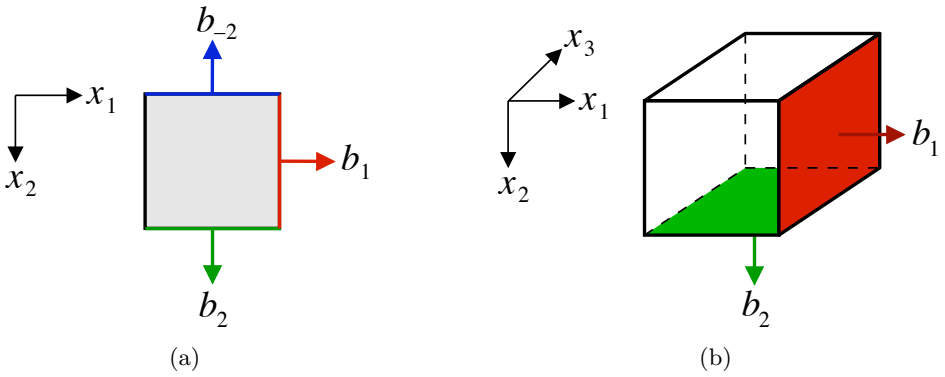


Figure 7.7: Bels are (a) oriented pixel edges in 2D and (b) oriented voxel faces in 3D.

7.1.4 Boundary of an Object

A set of bels separating an object and a co-object is defined as a *boundary* of the object. In Fig. 7.8, there are two objects (one in red and one in yellow) and two co-objects represented in pale blue. Between the red object, called *Object 1*, and the first connected component

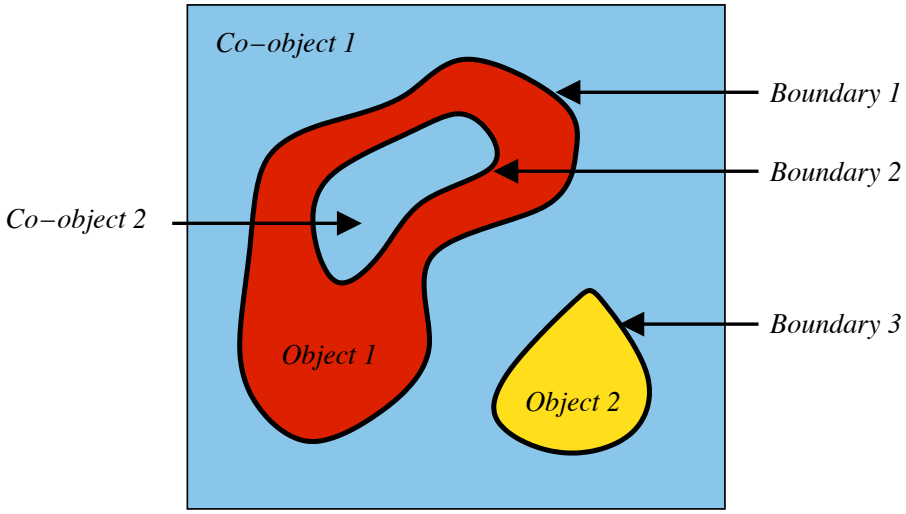


Figure 7.8: A boundary as a set of boundary elements separating an object and a co-object.

of the background, called *Co-object 1*, there is a boundary called *Boundary 1*. Similarly, between *Object 1* and *Co-object 2*, there is a boundary denoted as *Boundary 2*. For *Object 2*, we have only one boundary with respect to *Co-object 1*, which is *Boundary 3*.

More generally, with Υ and Φ denoting the object and background connectivity relations, let O be an object, and U be a co-object that is adjacent to O, meaning that there is an $o \in O$ and a $u \in U$ such that o and u are 1-adjacent. Then the *boundary surface* $S(O, U)$ separating O and U is the set of all bels (o, u) such that $o \in O$ and $u \in U$.

7.1.5 Adjacency of Bels

In addition to Υ and Φ-connectivity for object and co-object (background), there is also connectivity φ of the bels in a boundary curve (surface). Υ, Φ, and φ have to be chosen properly so that $S(O, U)$ satisfies certain desirable properties, such as:

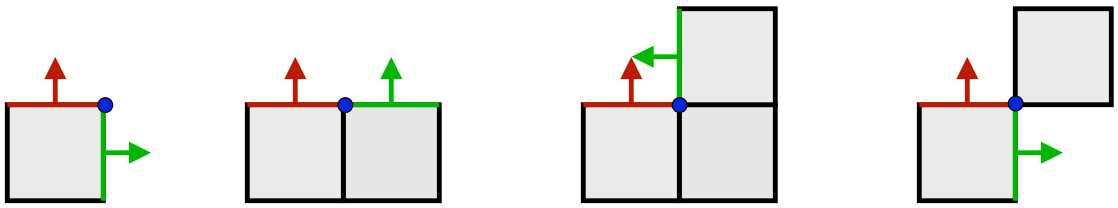
1. $S(O, U)$ is connected in the φ sense;

2. $S(O, U)$ is "closed" (a Jordan boundary);

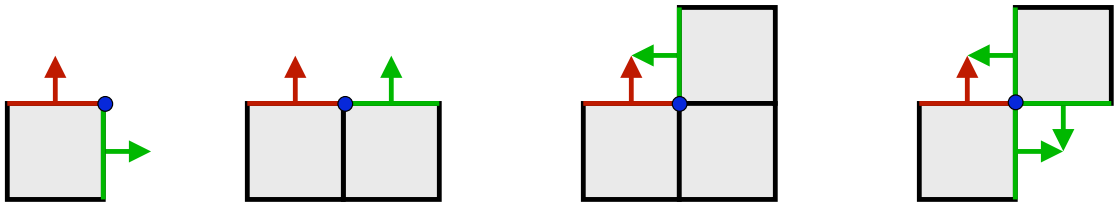
3. $S(O, U)$ is oriented.

Further, it should be possible to find $S(O, U)$ very efficiently, for any $\mathfrak{n} > 1$. The underlying theory and algorithms are described in [Udupa, 1994] in detail. Two possible triples (Υ, Φ, φ) are illustrated in Fig. 7.9 for $\mathfrak{n} = 2$. This figure illustrates the φ adjacency relation for different choices of Υ and Φ. Fig. 7.9(a) represents the adjacency for Υ = 4-connectivity and Φ = 8-connectivity, and Fig. 7.9(b) for Υ = 8-connectivity and Φ = 4-connectivity.

Then, as per φ, two edges are adjacent if they satisfy any of the four conditions shown in Fig. 7.9, with respect to the Υ and Φ used. We have represented the bel adjacent to the red edge, with respect to the blue node, as a green oriented edge. If two spels are not connected, then they will not have adjacent bels between them, as can be seen in the fourth condition presented in Fig. 7.9, where for Fig. 7.9(a), the red edge has an edge belonging to the same pixel adjacent to it at the blue vertex. In Fig. 7.9(b), the green edge adjacent to the red edge at the blue vertex has the b_{-1} orientation. Similarly, the other two green edges (with orientation b_1 and b_2) are adjacent to each other at the blue vertex.

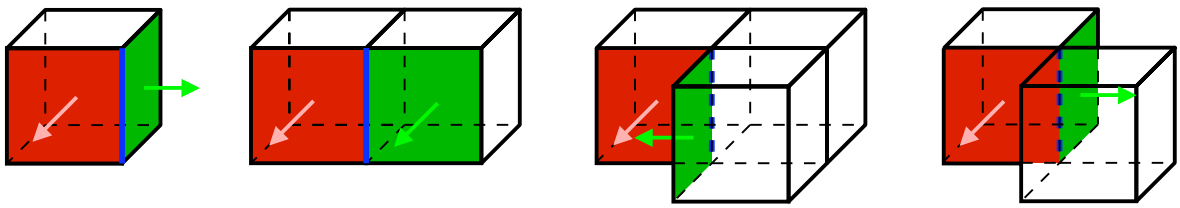


(a) Υ = 4-connectivity and Φ = 8-connectivity

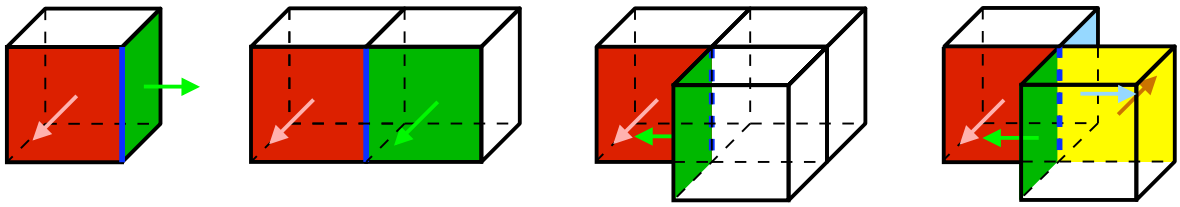


(b) Υ = 8-connectivity and Φ = 4-connectivity

Figure 7.9: Adjacency of bels in 2D.



(a) Υ = 6-connectivity and Φ = 18-connectivity



(b) Υ = 18-connectivity and Φ = 6-connectivity

Figure 7.10: Adjacency of bels in 3D.

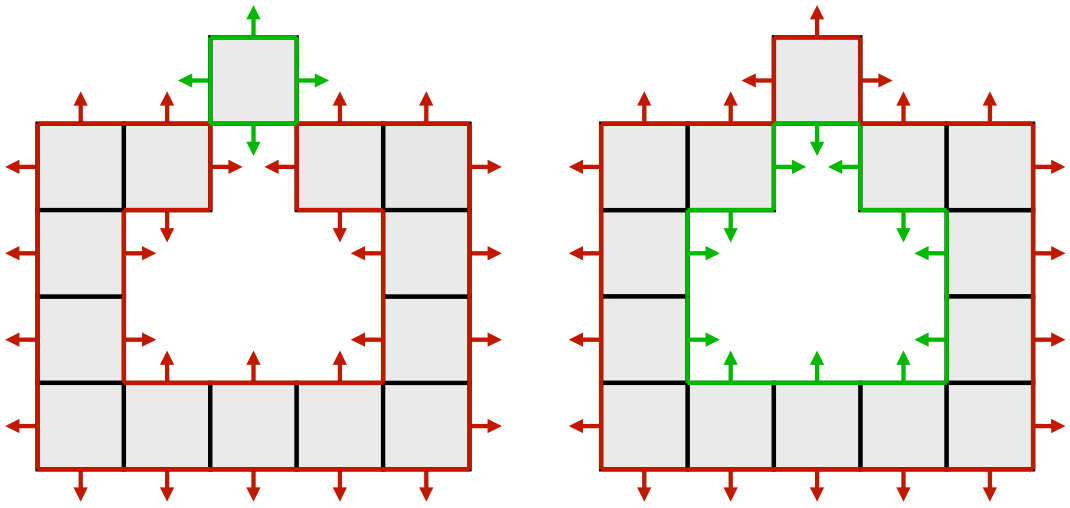
For the 3D case, similar configurations are shown in Fig. 7.10. As per φ, the green face is adjacent to the red face if they satisfy any of the four conditions shown in Fig. 7.10(a) for Υ = 6-connectivity and Φ = 18-connectivity, and Fig. 7.10(b) for Υ = 18-connectivity and Φ = 6-connectivity. Similarly to the 2D case, in Fig. 7.10(a), for the fourth condition, the red face has a green face belonging to the same voxel adjacent to it at the blue edge. In Fig. 7.10(b), the green face adjacent to the red face at the blue edge has the b_{-1} orientation. Similarly, the blue and yellow faces (with orientation b_1 and b_3) are adjacent to each other at the blue vertex.

Note that every face f in any 3D surface as defined above is adjacent to exactly 4 faces, one at each of the 4 edges of f. Any non empty surface $S(O, U)$ is connected, meaning, between any two faces f and f', there exists a sequence of faces $f_1, f_2, \ldots, f_m$, where $f = f_1$ and $f_m = f'$, such that f_i is adjacent to f_{i+1}, $1 \leq i < m$, and all faces f_i are in S.

7.1.6 Connectedness of Bels

Let e and l be two bels in a set H of bels. Then e is φ-connected to l in H if there exists a sequence of bels $e_1, e_2, \cdots, e_k$, all in H, such that $e_1 = e$ and $e_k = l$ and e_i is φ-adjacent to $e_i + 1$ for $i = 1, ..., k - 1$.

In Fig. 7.11, we have illustrated the connectedness of bels in the example given in Figs. 7.2-7.4, for the same settings of Υ and Φ used in Figs. 7.3 and 7.4. We notice that in both figures we have two oriented boundaries formed by connected bels, and in both cases the boundaries obtained are different, as they correspond to the connectivity defined by Υ and Φ. In Fig. 7.11(a), we have two independent objects with their corresponding connected set of bels represented in different colors for both objects. In Fig. 7.11(b), we have one object only, and the two sets of bels form two oriented boundaries, one internal boundary of the object and one external. The concept of connectivity will be extensively used in the RBS method to find connected fragments of the boundary (surface).



(a) Υ = 4-connectivity and Φ = 8-connectivity (b) Υ = 8-connectivity and Φ = 4-connectivity

Figure 7.11: Boundary connectedness.

7.2 RBS Principles and Overall Algorithm

In this section, we present a description of the principles under the 2D recursive boundary subdivision strategy and its overall algorithm.

7.2.1 Automatic Landmark Tagging Via 2D RBS

Let $\mathfrak{B} = \{\mathcal{B}_1, ..., \mathcal{B}_M\}$ be the set of boundaries of the training shapes. We assume each $\mathcal{B}_j, j = 1, ..., M$, to be an oriented connected and closed (Jordan) digital boundary [Udupa, 1994] as defined in the previous section. The method can be generalized easily from single Jordan boundary per object to multiple Jordan boundaries and in each of multiple objects. The method is based on recursive subdivision of each $\mathcal{B}_j$, first guided by the global shape and progressively by local shape. The hierarchical subdivision of each $\mathcal{B}_j$ generates a tree (Fig. 7.17), where each node denotes a boundary segment and arcs indicate subdivision. There is a 1:1 correspondence among the trees for all $\mathcal{B}_j$s, and corresponding nodes in the different trees represent boundary segments of similar shape in the different $\mathcal{B}_j$s.

We assume that the boundaries in $\mathfrak{B}$ have been aligned via translation, scaling, and rotation as commonly done in most modelling methods. In the following, we will illustrate the steps of this approach in a training set of two shapes, presented in Fig. 7.12.

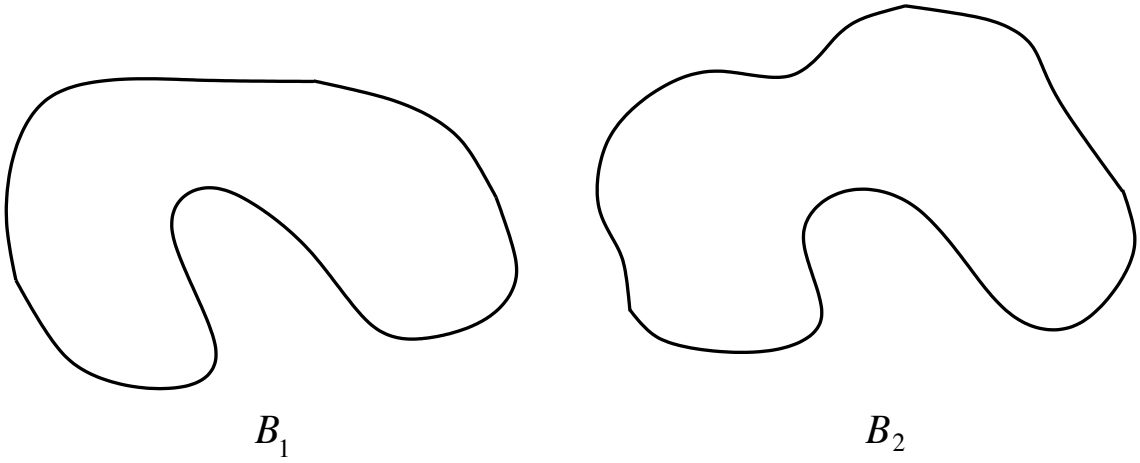


Figure 7.12: A training set of two shapes $\mathcal{B}_1$ and $\mathcal{B}_2$.

Suppose we have a method α to first find the two most prominent points p_1, p_2 on each $\mathcal{B}_j$. For example, p_1 and p_2 are the farthest points on $\mathcal{B}_j$, as shown in Fig. 7.13 for the shapes in Fig. 7.12. They may also be found by any other criteria (e.g. registration). The

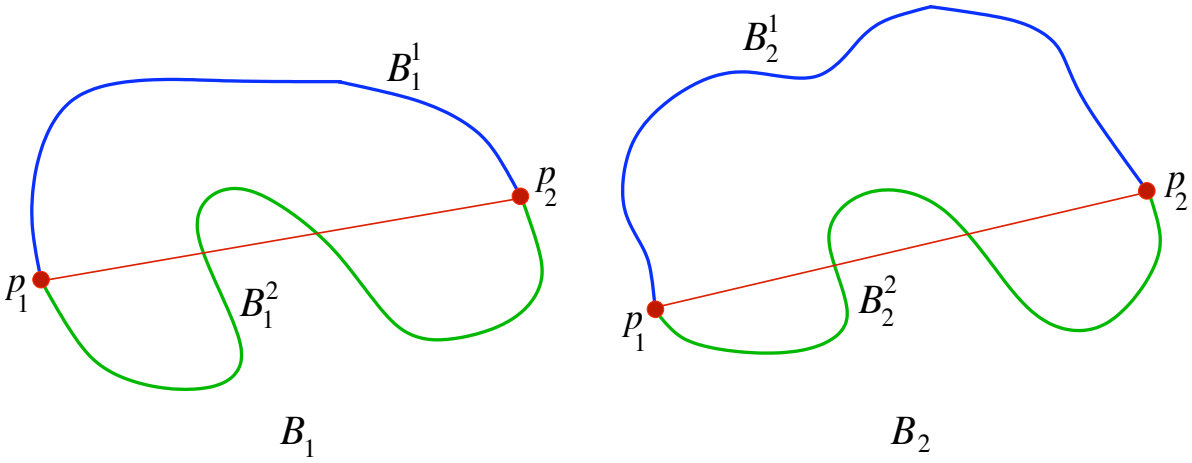


Figure 7.13: Selection of the initial landmarks through method α on the training set in Fig. 7.12. In this case, method α finds the two farthest points p_1 and p_2 on each training shape.

points p_1 and p_2 define two connected segments on each $\mathcal{B}_j$ denoted $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$, representing respectively the upper and lower connected pieces from p_1 to p_2.

Suppose also that we have another method β to find unique points p_3 on each $\mathcal{B}_j^1$ and p_4 on each $\mathcal{B}_j^2$. Different approaches for method β are described in the following section. Once we find p_3 in all $\mathcal{B}_j$ as per method β, we have new lines p_1p_3 and p_3p_2, and the associated points divide each $\mathcal{B}_j^1$ into segments $\mathcal{B}_j^{11}$ and $\mathcal{B}_j^{12}$. This is illustrated in Fig. 7.14. Corresponding segments among shapes are represented with the same color. Similarly, each $\mathcal{B}_j^2$

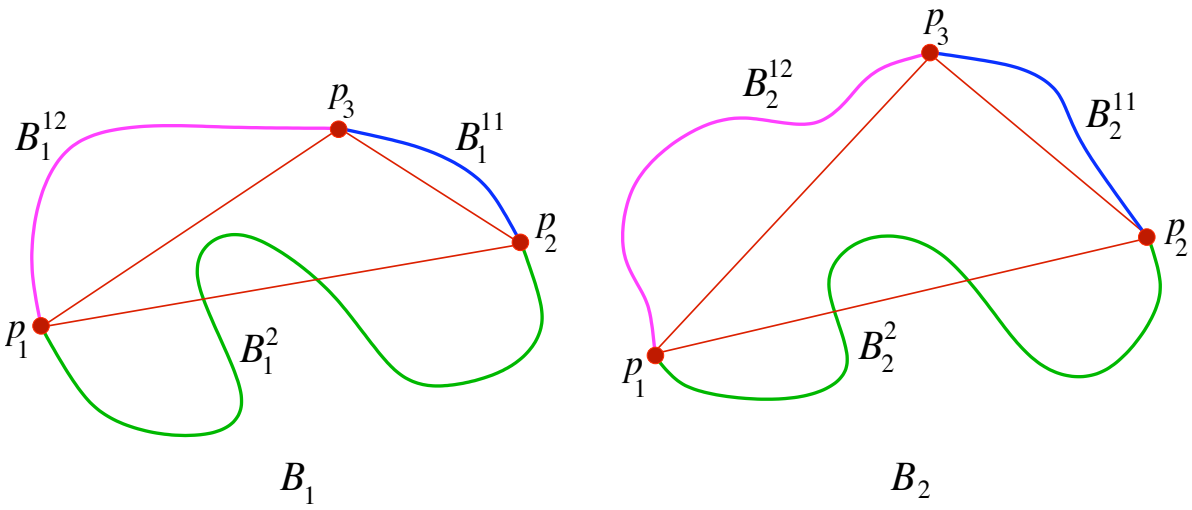


Figure 7.14: Subdivision of $\mathcal{B}_j^1$ into two new segments for the shapes in Fig. 7.12.

will be divided into segments $\mathcal{B}_j^{21}$ and $\mathcal{B}_j^{22}$.

If each of these four new segments is flat enough, as determined by a parameter δ, subdivision ceases, else, it continues by using method β to find new points, eventually yielding points $p_1, p_2, ..., p_n$. In Fig. 7.15, we illustrate all the subdivisions produced for a certain δ for the shapes in Fig. 7.12. In this figure, the red lines denote all the lines involved in the recursive subdivision process for that example. Note that when p_kp_l is being considered on $\mathcal{B}_j$ for subdivision, the corresponding p_kp_l is considered on all $\mathcal{B}_j$s in $\mathfrak{B}$. Therefore, we can notice the same pattern of lines appearing on both shapes in Fig. 7.15. The dotted lines

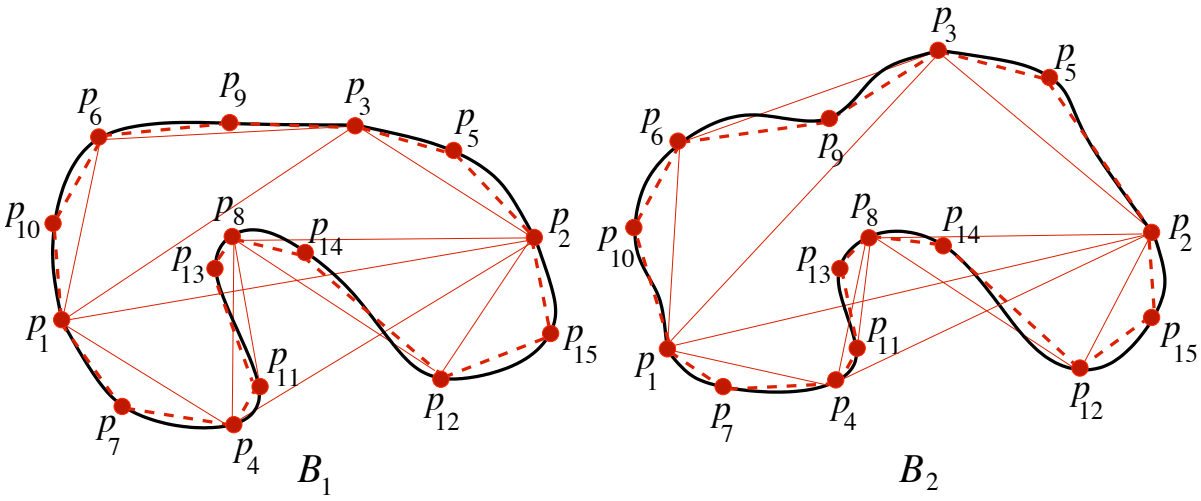


Figure 7.15: Simultaneous recursive boundary subdivision of shapes to find landmarks for the shapes in Fig. 7.12.

indicate the last subdivision considered that satisfied δ. During subdivision, even if there is one boundary $\mathcal{B}_h$ in $\mathfrak{B}$ for which a segment X satisfies division criterion δ, the segment corresponding to X is subdivided in all $\mathcal{B}_j$s in $\mathfrak{B}$. Therefore, the final polygonal representation of all boundaries, with the same number of corresponding landmarks, is represented by dotted lines in Fig. 7.16, and contains all the most prominent landmarks existing among shapes of the training set. The level of detail obtained depends on the selection of the parameter δ.

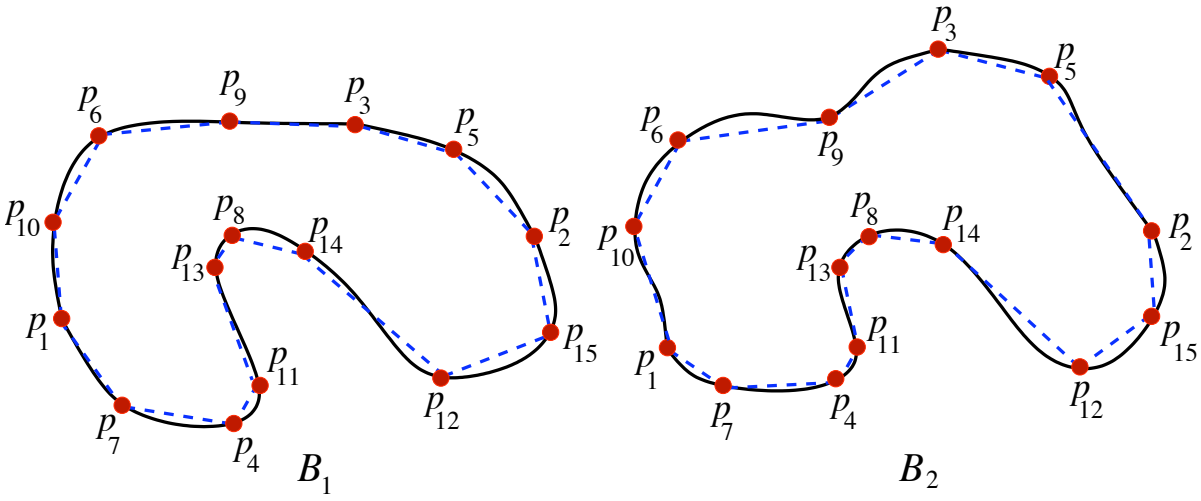


Figure 7.16: Landmarks and polygonal representations for the shapes in Fig. 7.12.

The boundary subdivision tree corresponding to the shapes in Fig. 7.12 is portrayed in Fig. 7.17. The tree is the same for all $\mathcal{B}_j$, as it characterizes the subdivision process, which

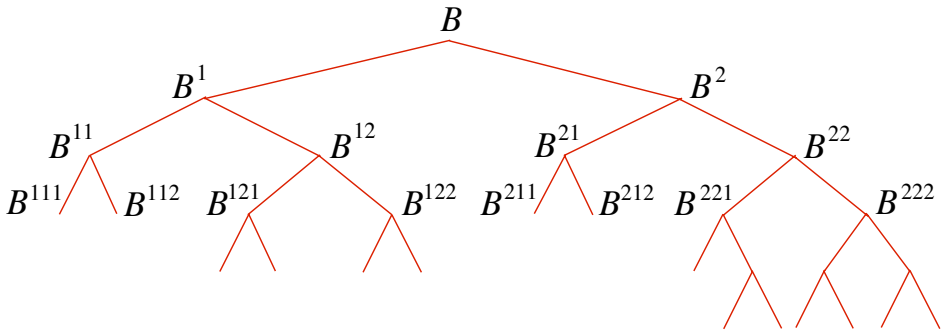


Figure 7.17: Boundary subdivision tree contains global-to-local shape information and shape correspondence information among training object shapes. $\mathcal{B}_j$ is replaced by $\mathcal{B}$ in the tree for simplicity.

is the same for all shapes in a training set. The tree has rich global-to-local information and is useful in global to local shape analysis, change detection, etc. The most prominent landmarks detected correspond to the segments at the top of the tree, whereas the less prominent, selected locally at a finer subdivision level, are at the bottom of the tree.

Overall, the concept of connectivity is very important, as it guarantees the correspondence of segments among shapes at each subdivision step. If the connectivity is not taken into account, and the selection of segments is made with respect to the cutting line at each iteration, then, the correspondence is lost and undesirable results may appear as the approach becomes susceptible to small shape changes. This is why it is important to assure the connectivity at each stage to find corresponding landmarks as a result.

7.2.2 Overall Algorithm

In the following, we present the 2D RBS general algorithm.

Algorithm 7.1: **2D RBS** - Recursive Boundary Subdivision

Input: Boundaries $\mathcal{B}_j : j = 1, ..., M$; parameter δ; methods α and β.

Output: Landmarks $p_1^j, ..., p_n^j, j = 1, ..., M$.

Auxiliary Data: Queue Q_j, associated with each boundary $\mathcal{B}_j, j = 1, ..., M$.

Initialization

for $j = 1, ..., M$ **do**

1. Call method α to find the initial two landmarks p_1^j, p_2^j on $\mathcal{B}_j$. These points define a line $p_1^j p_2^j$ that divides $\mathcal{B}_j$ into two connected segments $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$. Each segment is connected to p_1^j and p_2^j.
2. Enqueue the two boundary segments $\mathcal{B}_j^X$ in Q_j associated with $\mathcal{B}_j$. Note the pair $< p_1^j, p_2^j >$ associated with these segments.

end

Recursive Subdivision

while Q_j, $j = 1, ..., M$, *is not empty* **do**

3. Dequeue boundary segment $\mathcal{B}_j^X$ from Q_j for each $j = 1, ..., M$. Using the associated point pair $< p_k^j, p_l^j >$, determine if $\mathcal{B}_j^X$ needs to be further subdivided as per method β for each $j = 1, ..., M$.
4. If there is some i in $\{1, ..., M\}$ such that $\mathcal{B}_i^X$ can be subdivided, then, for each $j = 1, ..., M$, determine the new segment of $\mathcal{B}_j^X$ using $< p_k^j, p_l^j >$, and enqueue the segment in Q_j.

end

5. Output $p_1^j, ..., p_n^j, j = 1, ..., M$.

7.3 Boundary Landmark Selection Strategies

In this section, we propose different strategies for methods α and β, and for the criterion used for parameter δ.

7.3.1 Selection Based on Distance

Methods α and β can be selected based on distance. Method α can be as simple as selecting the two farthest points on the boundary $\mathcal{B}_j$.

Method β can be the farthest distance from the line associated with the segment in $\mathcal{B}_j$. This distance has to be bigger than a threshold δ, which is the stopping criterion for the method. In Fig. 7.18, β is considered to base its decision on the farthest distance from the line p_1p_2 to the boundary on each of $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$. In Fig. 7.18, the point p_3 that will lead

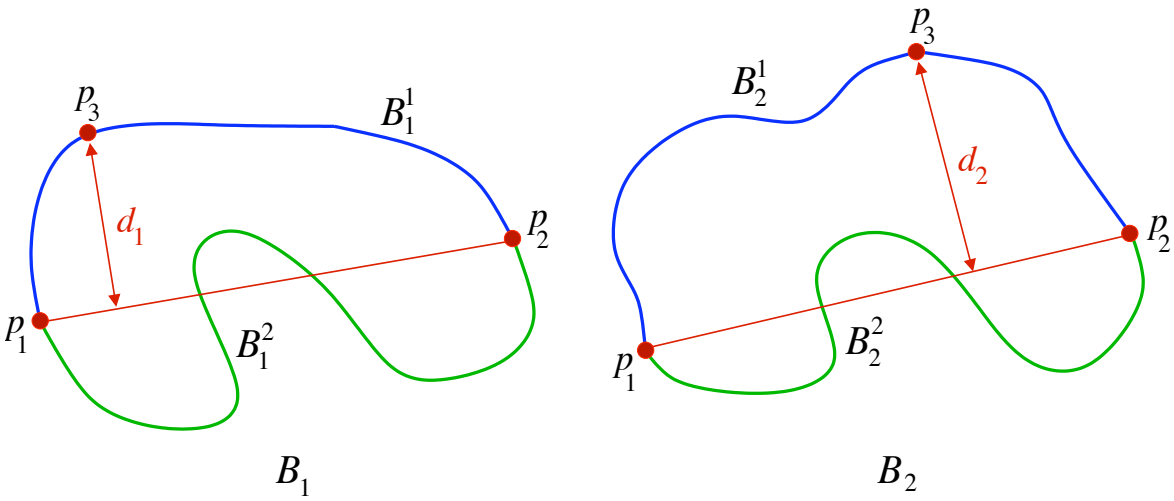


Figure 7.18: Selecting new landmarks during recursive subdivision through method β.

the subdivision is the one belonging to $\mathcal{B}_2^1$, because the distance d_2 from the line p_1p_2 to the segment $\mathcal{B}_2^1$ is greater than the distance d_1 from the line p_1p_2 to the segment $\mathcal{B}_1^1$. Therefore, the point p_3 in $\mathcal{B}_2$ will guide the subdivision of the corresponding segments $\mathcal{B}_1^1$ and $\mathcal{B}_2^1$, and it will be propagated to $\mathcal{B}_1^1$ (for example, via closest point propagation, as explained in Section 5.5) to find the corresponding location of p_3 in $\mathcal{B}_1$, as shown in Fig. 7.19. In general, for this choice of method β, if p_3 is found in $\mathcal{B}_k$, it will lead the subdivision of all corresponding segments, and its location will be propagated to all shapes $\mathcal{B}_j$, $j \neq k$.

δ is in this case the distance of the segment to the line generating that segment and is

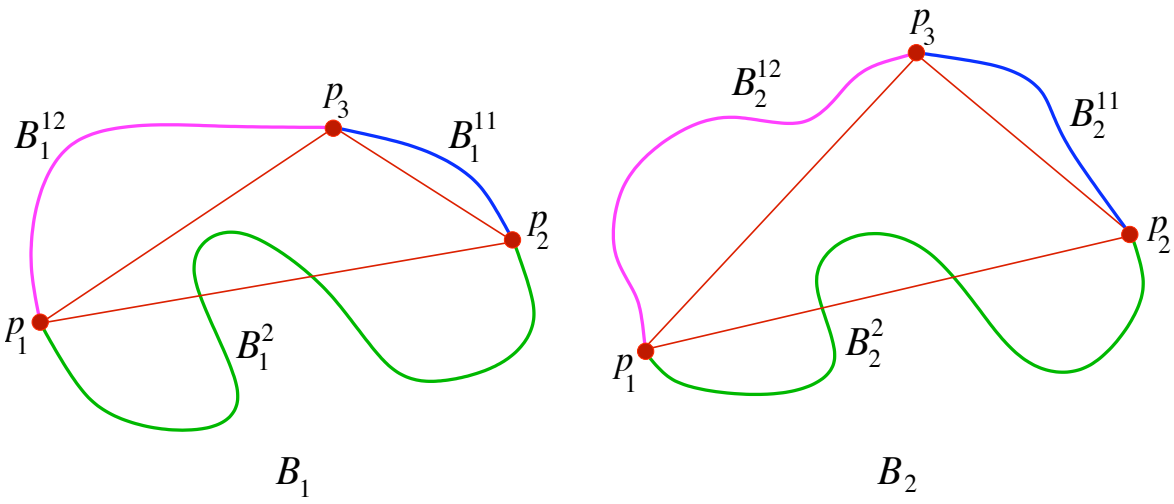


Figure 7.19: Selection of p_3 based on the farthest distance from the line p_1p_2 to the segment $\mathcal{B}_1^1$ for Fig. 7.18.

given as a threshold. The recursive subdivision will stop when no segment presents a distance, from its associated pair to the segment, greater than δ. Other options are also possible.

This approach relies on the accuracy of the registration used to align all shapes of the training set, since some landmark propagation is necessary in some cases, and the location of these landmarks will be better if the registration performed is good.

7.3.2 Selection Based on Principal Component Analysis

As an alternative to distance-based approaches, we propose the selection of landmarks guided by Principal Component Analysis (PCA) of the boundary segments for methods α, β. PCA aims at reducing the dimensionality of a set of points, by representing them with respect to the two directions in which the variance is maximum. Let $\mathbf{x} = (x_1, y_1, x_2, y_2, ..., x_q, y_q)$ be the vector representing the (x, y) coordinates of the q points comprising any boundary/boundary segment $\mathcal{B}$. Then, the PCA of $\mathcal{B}$ yields a principal axes system for $\mathcal{B}$ comprising of the geometric centroid $G = (\overline{x}, \overline{y})$ of $\mathcal{B}$, where $(\overline{x}, \overline{y})$ is the mean location of the q points of $\mathcal{B}$, and two orthogonal axes PA_1 and PA_2, where PA_1 represents the major inertia axis of $\mathcal{B}$ and PA_2 represents the second inertia axis. In the following, we will refer to the first eigenvector

associated to with PA_1 as V_1 and the second as V_2. Their corresponding eigenvalues will be denoted by λ_1 and λ_2, respectively. Let G_j^X be the centroid to the boundary segment $\mathcal{B}_j^X$.

We define method α as finding initial points p_1 and p_2 via PCA, as the outermost points on $\mathcal{B}_j$ along the primary principal axis PA_1, of $\mathcal{B}_j$, $j = 1, ..., M$. In Fig. 7.20 we have represented the initialization step by using the first principal axis to define p_1 and p_2 for the boundaries in Fig. 7.12. We need to guarantee that the order of selection of p_1 and p_2 on all

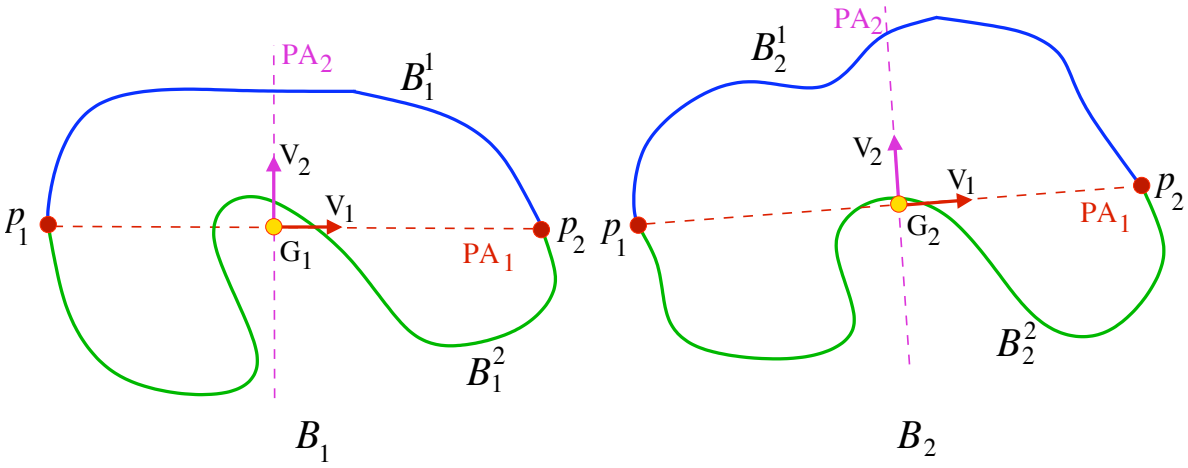


Figure 7.20: Selection of p_1 and p_2 by method α based on PCA.

shapes is the same. The first eigenvector associated with a boundary $\mathcal{B}_j$ is not necessarily oriented in equivalent directions for all shapes. Therefore, we use the eigenvector V_1 associated with $\mathcal{B}_1$ to find equivalent orientations of V_1 among shapes, so all p_1 and p_2 among shapes of the training set are taken in the same order as in $\mathcal{B}_1$.

Method β will then find new landmarks p_k in a boundary segment $\mathcal{B}_j^X$ in the direction of the second principal axis PA_2, passing through the geometric center of the segment G_j^X, as shown in Figs. 7.21 and 7.22. In this case, we take the point p_3 to be the point where PA_2 and the boundary segment intersect. We do not have to take care of the order as for method α, since here we will detect only one landmark in the boundary segment. Point p_3 is found using this strategy on all $\mathcal{B}_j$. Now we have new lines p_1p_3 and p_3p_2, which divide

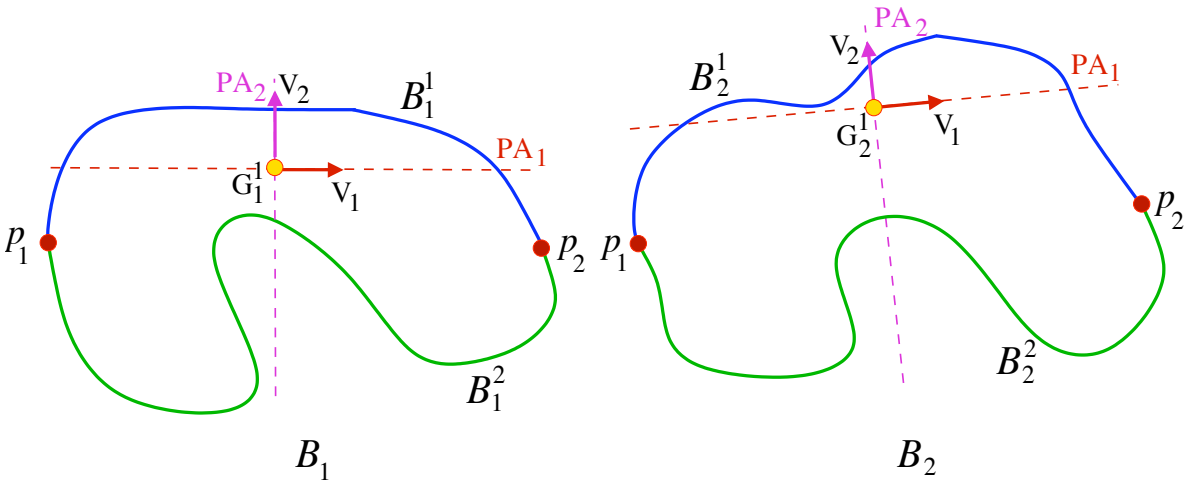


Figure 7.21: PCA corresponding to $\mathcal{B}_j^1$.

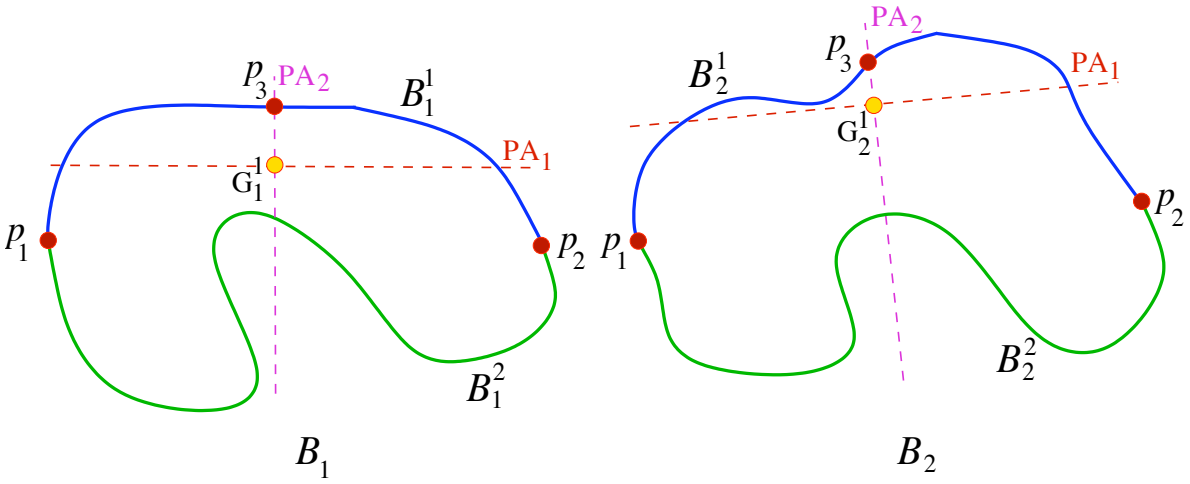


Figure 7.22: Selection of p_3 guided by PA_2 for the boundary segments $\mathcal{B}_j^1$.

each $\mathcal{B}_j^1$ into segments $\mathcal{B}_j^{11}$ and $\mathcal{B}_j^{12}$, as illustrated in Fig. 7.23. Corresponding segments are

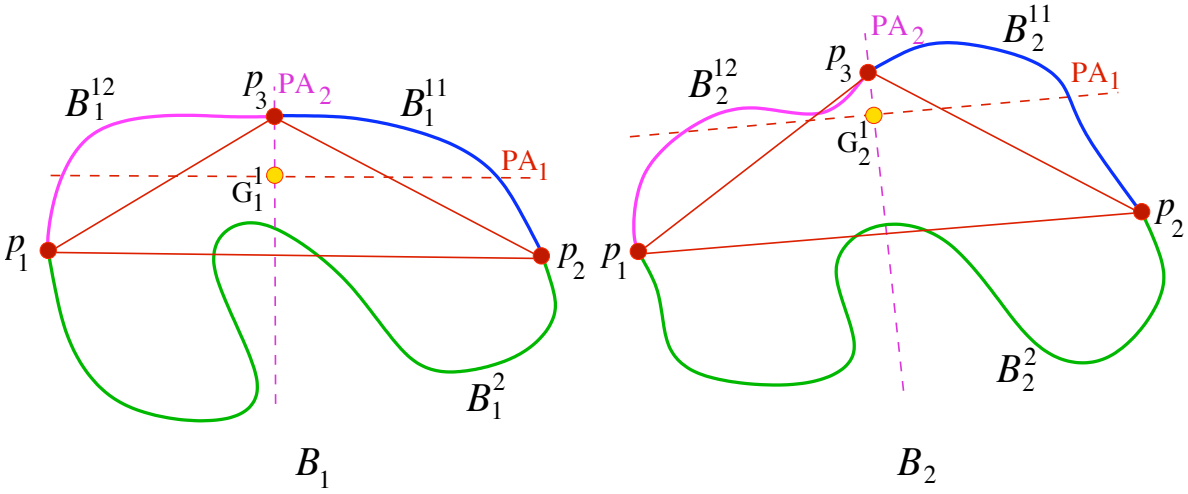


Figure 7.23: Subdivision of $\mathcal{B}_j^1$ into segments $\mathcal{B}_j^{11}$ and $\mathcal{B}_j^{12}$.

represented in the same color. We take directly the landmarks obtained by intersection of the segment and their corresponding PA_2, no propagation of landmarks is involved in this approach.

In this case, the parameter δ is the eigenvalue λ_2 associated with the second inertia axis, and λ_2 can be used to decide if subdivision should continue. Even if only one eigenvalue among all shapes is greater than δ, the subdivision is applied to all corresponding segments.

A principal axis might not pass exactly through an existing bel in the boundary. Therefore, we select as landmark the bel closest to the principal axis of interest in each case. To find this bel efficiently, we measure all distances between consecutive bels in the boundary. If the distance from the principal axis to any bel in the boundary is smaller than the maximum distance value between successive bels, then, we mark them as intersecting bels. From this cluster of intersecting bels, we select as landmark the one that is farthest from the line connecting the two landmarks generating that segment.

An interesting attribute of this approach is that it does not require for the shapes to be

registered a priori, as there is no landmark propagation involved and corresponding landmarks are directly found on the boundaries via PCA of corresponding shapes. In Fig. 7.24, we show how the correspondence of landmarks is when using this approach with $\delta = 2.5$ for the training set of shapes presented in Fig. 7.12.

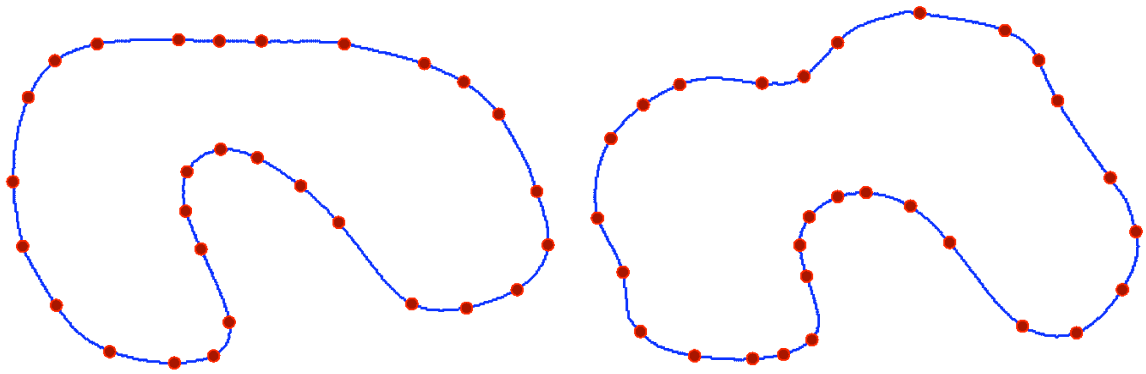


Figure 7.24: RBS based on PCA with $\delta = 2.5$ for the training set of shapes in Fig. 7.12.

As we can observe, this strategy based on PCA detects corresponding landmarks that portray shape features present among shapes in the training set, and not necessarily present in all the shapes. Even if the shape feature exists for only one particular shape, it is automatically incorporated in all the shapes of the training set. For example, in Fig. 7.24 at the top part of the shape on the right there is an indentation that is well captured by the principal axes at different subdivisions. The information in this area is also incorporated in shape on the left even if the indentation does not appear in that shape, establishing the correspondence between landmarks as best as possible that seems intuitively correct.

7.4 Virtual Landmarks

Until now, in all the landmark tagging approaches studied in this thesis and in the literature, landmarks were confined to the boundary of each shape at corresponding locations among shapes. However, a landmark may also be defined as any point in the space where the shape is defined, but not necessarily on it, such that the landmark has a consistent relationship with the shape boundary. In actuality, landmarks computationally defined on the boundary

may not even correspond to any real anatomic landmark. In this section, we generalize the concept of landmark originating from an anatomic motivation, and hence defined on boundaries, to a more general setting by letting them be anywhere inside or outside the object boundary. Therefore, we introduce the novel concept of virtual landmarks. The main idea is that they will have a consistent relationship with the shape boundary, making the notion more general and not forcing the landmarks to be confined to the boundary.

In this section, we present two approaches for virtual landmark tagging, both based on the PCA approach previously described. The main characteristic of this family of methods is that they will subdivide the boundaries as explained in the PCA-based approach, but will define as landmarks points that are not included in the boundary shapes but that are otherwise related to the shape. The reason why we use points on the boundary for subdivision of the boundary shapes is that they help subdividing shapes consistently, whereas points considered outside or inside the boundary can give rise to many possible scenarios for subdivision, which depend on the location of the point, and which may not be consistent among shapes. Taking the point on the boundary for the subdivision of segments guarantees that the process undertaken is the same and consistent at each stage.

7.4.1 Selection Based on Centroids

The first approach makes use of the centroids obtained at each subdivision in the RBS approach. In the following, we will refer to the cut points on the boundary as C_k, $k = 1, ..., n+1$, and to the virtual landmarks as p_i, $i = 1, ..., n$, for each boundary $\mathcal{B}_j$. In this approach, method α will select only the first landmark p_1, in all $\mathcal{B}_j$, as the centroid of each boundary shape. Two cut points C_1 and C_2 are defined at the intersection of the first principal axis with each boundary for all $\mathcal{B}_j$. In Fig. 7.25, we can see the result of applying method α to initialize the subdivision process. The line C_1C_2 resulting from cutting each boundary $\mathcal{B}_j$ by the first principal axis PA_1 is used to first subdivide each $\mathcal{B}_j$ into two connected pieces $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$. The first virtual landmark p_1 is located at this step and corresponds to the centroid of each $\mathcal{B}_j$. Note that the virtual landmarks are not used for the subdivision of the boundary

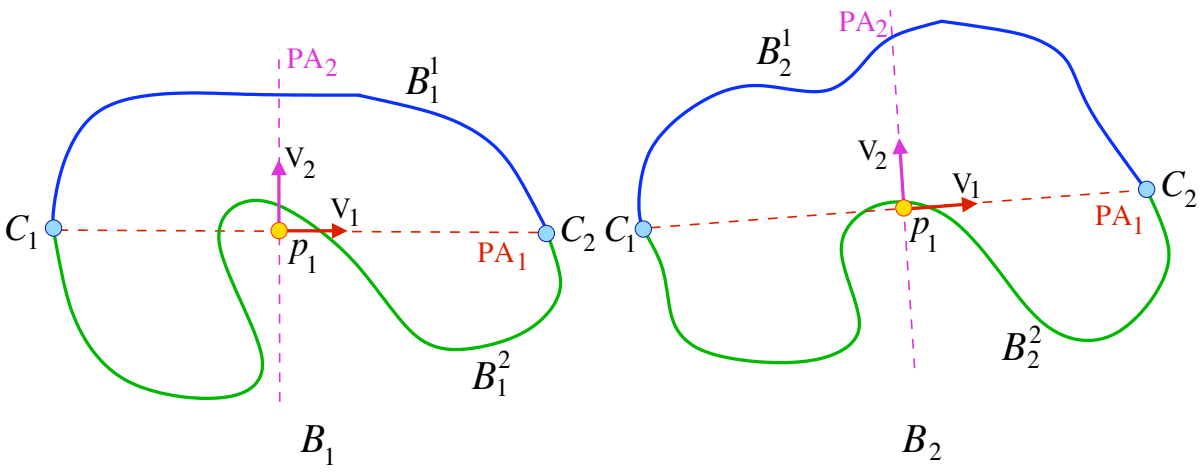


Figure 7.25: Method α for the selection of the first virtual landmark p_1 for the training set of shapes in Fig. 7.12.

shapes at any stage of the method.

Method β will then be called subsequently to find new virtual landmarks p_i, $i = 2, ..., n$, in successive boundary segments $\mathcal{B}_j^X$, at the geometric center of each segment, as shown in Fig. 7.26. The intersection between the second principal axis PA_2 and the segment defines a

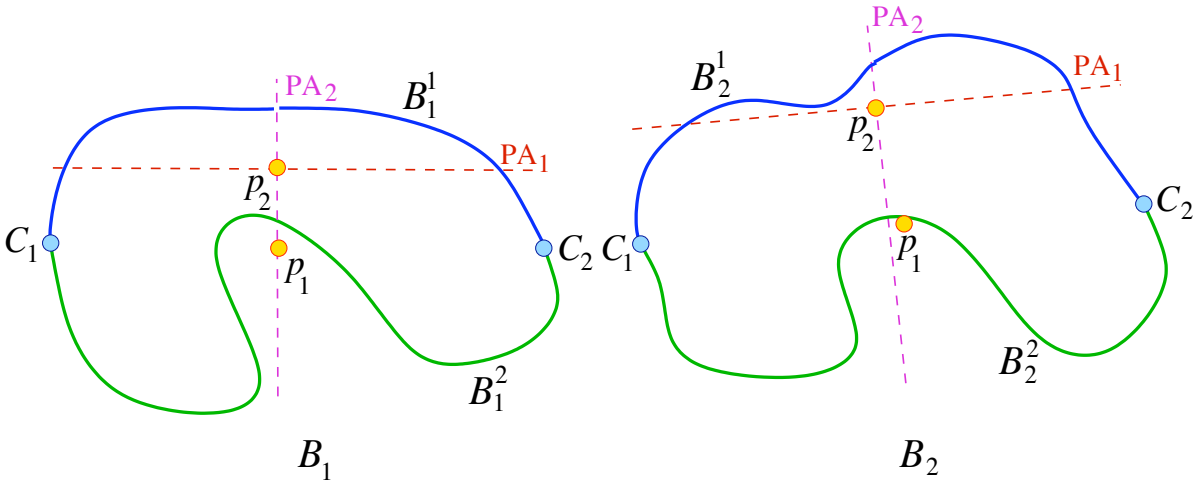


Figure 7.26: Method β for the selection of a new landmark p_2 in a boundary segment $\mathcal{B}_j^1$ for the training set of shapes in Fig. 7.12.

new subdivision point, as shown in Fig. 7.27, where C_3 is the intersecting point between $\mathcal{B}_j^1$

and the associated PA_2. The point C_3 subdivides $\mathcal{B}_j^1$ into two connected segments $\mathcal{B}_j^{11}$ and

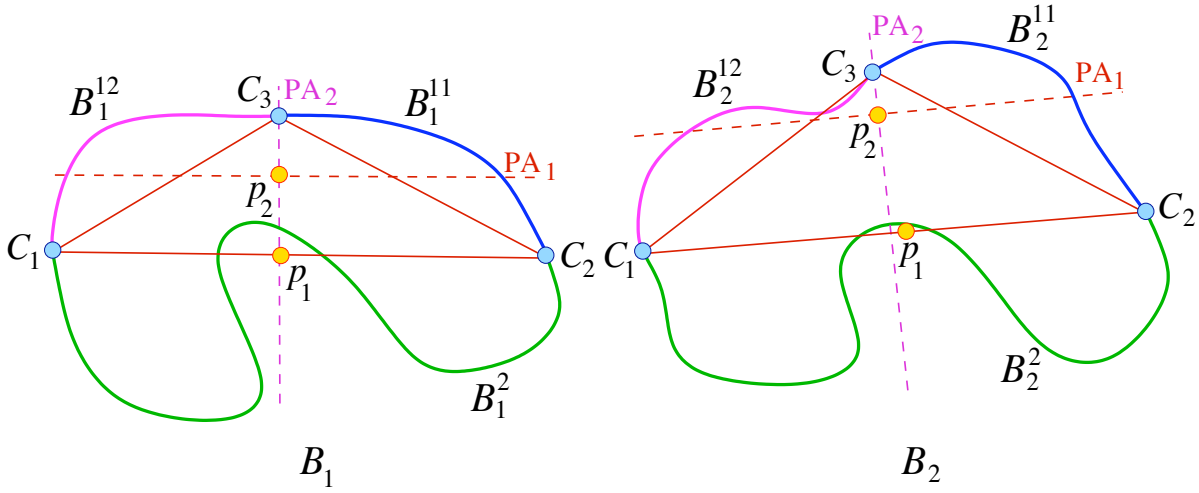


Figure 7.27: Subdivision of the segments $\mathcal{B}_j^1$ for the training set of shapes in Fig. 7.12.

$\mathcal{B}_j^{12}$.

Similarly to the PCA-based RBS approach, the subdivision will continue until the second eigenvalue associated with each segment is greater than a parameter δ even at one boundary segment $\mathcal{B}_j^X$. In Fig. 7.28, we can see the result of applying the method for $\delta = 4$. We have

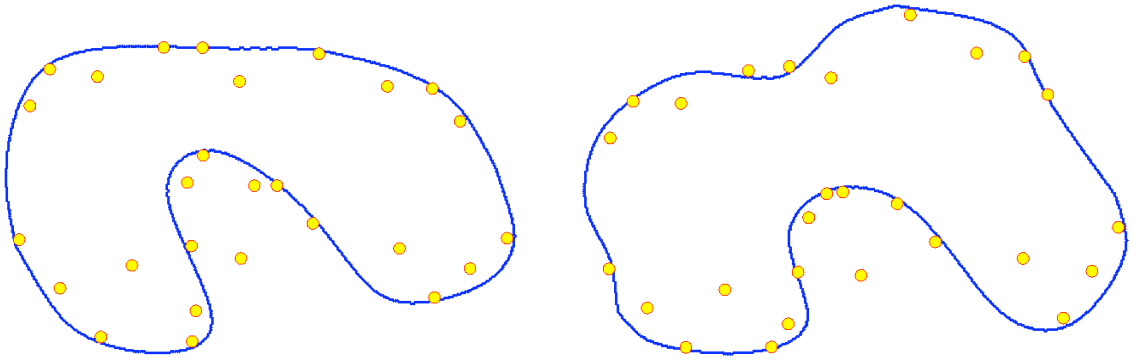


Figure 7.28: Virtual landmarks obtained with $\delta = 4$ for the training set of shapes in Fig. 7.12.

represented virtual landmarks with a yellow dot for shapes in Fig. 7.12. We can see that the centroids located on both shapes, at different stages, are corresponding and seem robust. At early stages of the approach, the centroids are located in the center of the shapes and they

move towards the boundary in a progressive manner at latter stages, establishing correspondence of landmarks from a global manner initially to local levels around the boundary in later stages.

7.4.2 Selection Based on Eigenvalues

In this variant of the PCA approach, our goal is to locate virtual landmarks on the principal axes associated with each segment, and at a distance proportional to the eigenvalue at each stage. The direction in which we place the landmarks is defined per convention, and it is assumed to be always the direction pointing from the inside to the outside of each shape. This way, we guarantee that the selection of landmarks is always attained by using the same criteria among shapes. The problem of using the direction directly obtained by the eigenvectors is that these are not always consistent and do not necessarily point towards the outside of the segment for all shapes.

To find the direction pointing out of the shape, we can make use of the centroids at each stage. However, for the result to be consistent, we need to keep track of where the centroid is. The centroid can be located inside or outside the boundary, varying among shapes, since it depends on global and local shape characteristics. Therefore, to be consistent in the selection of landmarks and maintain the correspondence of landmarks among shapes, we need a robust approach to place landmarks, independently of the shape considered. We will identify where the centroid of each segment lies with respect to the boundary, and depending on its location, locate the landmark following the eigenvector in the direction pointing to the outside of the shape boundary from the centroid. To determine the direction in which we want the eigenvector to point, we apply a Euclidean signed distance transform (cf. Section 5.3.2) on each shape of the training set. This way, we obtain negative distance values inside the shape and positive outside, with respect to the boundary of each shape. The sign of the centroid is obtained by examining its location in the Euclidean distance map. At each subdivision, the sign tells us if the centroid is inside or outside the shape, and therefore will help us in deciding the direction in which the new virtual landmark needs to be placed. If the cen-

troid is inside the shape, then the associated eigenvector selected should point towards the boundary, whereas, if it is outside the shape, it should point in the direction going away from the boundary. That way, we assure the correspondence of landmarks among shapes of the training set, since their location will always be in the same direction relative to the shape boundary.

As for the other approaches of this family, method α will be guided by the first principal axis PA_1 of each boundary shape $\mathcal{B}_j$. Two virtual landmarks p_1 and p_2 are selected at each side of the centroid, on PA_1, and at a distance $a\lambda_1$, where a is a constant and λ_1 is the eigenvalue associated with the first eigenvector in the direction of PA_1 for each $\mathcal{B}_j$. This step is illustrated in Fig. 7.29. The order in which we select p_1 and p_2 should be the same

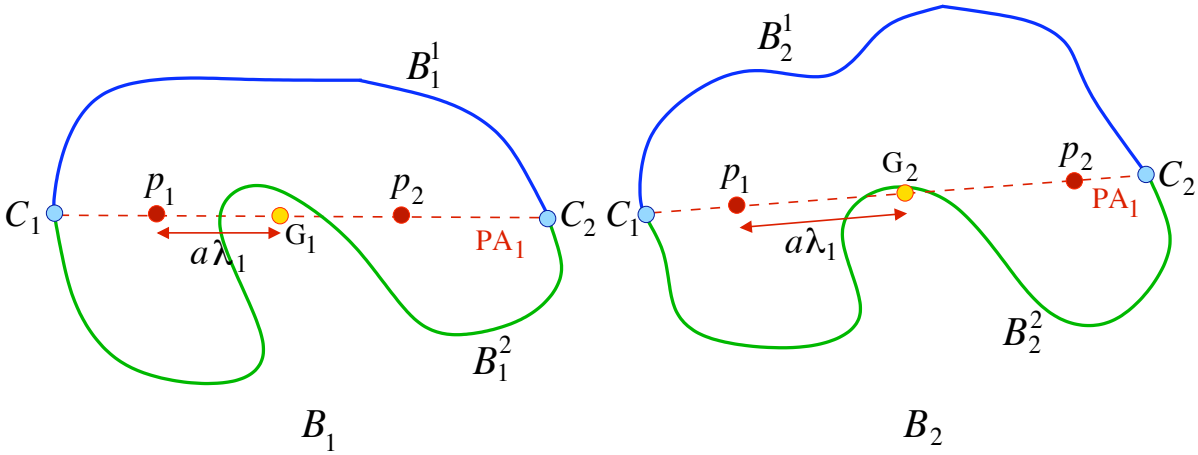


Figure 7.29: Method α for the selection of the virtual landmarks p_1 and p_2, at a distance $a\lambda_1$ from the centroid G_j for the training set of shapes $\mathcal{B}_j$ in Fig. 7.12.

among shapes to establish the correct correspondence. This can be assigned manually or automatically by selecting the direction of each landmark in a similar way among shapes of the training set. The boundary $\mathcal{B}_j$ is subdivided at the cut points C_1 and C_2, resulting from intersecting the boundary with the first principal axis PA_1 at each side of the boundary shape. The subdivision generates two connected segments $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$ for each $\mathcal{B}_j$.

By method β we select a new landmark on a connected boundary segment by using the

second principal axis PA_2 associated with that segment. The new landmark is placed at a distance $a\lambda_2$ from the centroid and in the direction going from the inside to the outside of each shape. This step is illustrated in Fig. 7.30. Note that the virtual landmarks will all be

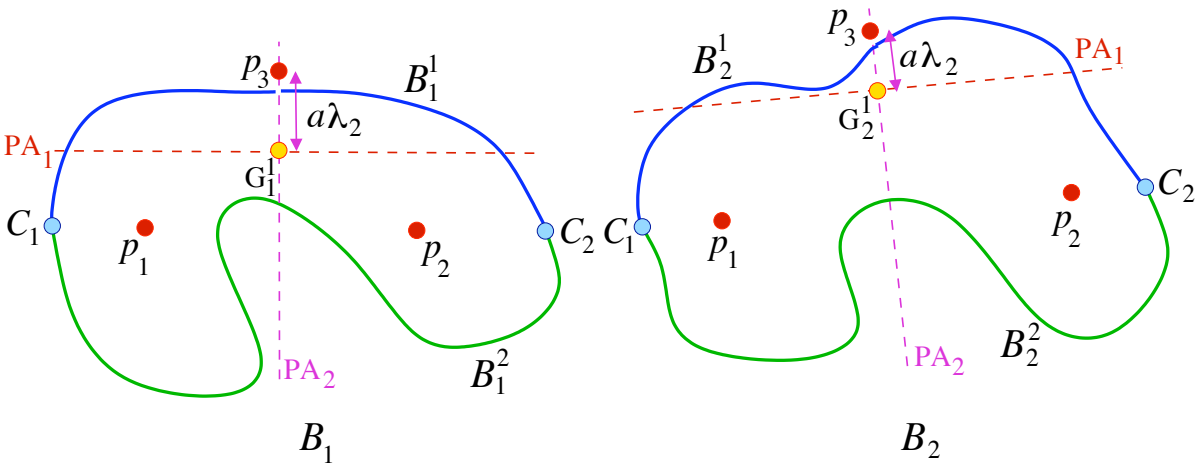


Figure 7.30: Method β for the selection of the virtual landmarks p_3, at a distance $a\lambda_2$ of the centroid G_j^1 for the segments $\mathcal{B}_j^1$ for the training set of shapes in Fig. 7.12.

consistently located in the same direction pointing towards the outside of each shape, and they can fall inside or outside the boundary, depending on the characteristics of the shapes considered. It is not required for them to fall all inside or all outside among shapes, since the correspondence is guaranteed by selecting the same direction for its location among shapes.

After detecting a new landmark, the subdivision of each resulting segment $\mathcal{B}_j^1$ and $\mathcal{B}_j^2$ is performed at the intersecting point between the second axis and each segment. This point will generate new lines that will subdivide the segment into two new segments as shown in Fig. 7.31. In this case, the cut point C_3 generates two new subdivision lines C_1C_3 and C_3C_2, which divide $\mathcal{B}_j^1$ into two connected segments $\mathcal{B}_j^{11}$ and $\mathcal{B}_j^{12}$. The subdivision will continue until there is no segment with an associated eigenvalue greater than a parameter δ, which in this case consists of the second eigenvalue associated with the segment under consideration.

In this strategy, we can use the parameters δ and a to control the level of detail and the

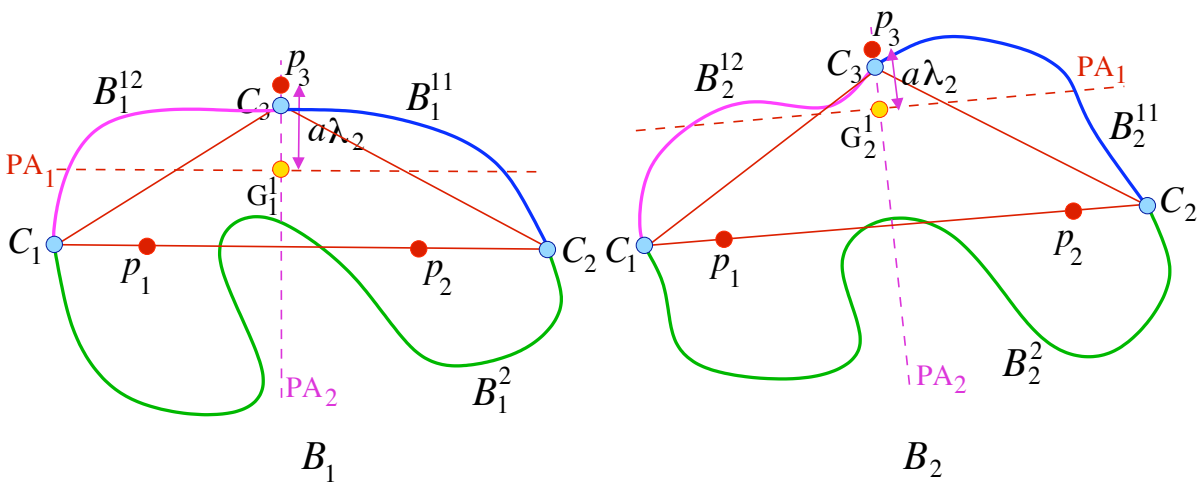


Figure 7.31: Subdivision of the segments $\mathcal{B}_j^1$ by using C_3 for the training set of shapes in Fig. 7.12.

location of the landmarks. By varying the parameter a, we obtain landmarks at different corresponding positions. In Fig. 7.32, we have represented the results obtained for $\delta = 10$ and $a = 1$. The centroids are represented in yellow and the virtual landmarks in red. As we can observe, all the virtual landmarks are directed to the outside of the shape on a line passing through them and their corresponding centroids. Both, the centroids and the virtual

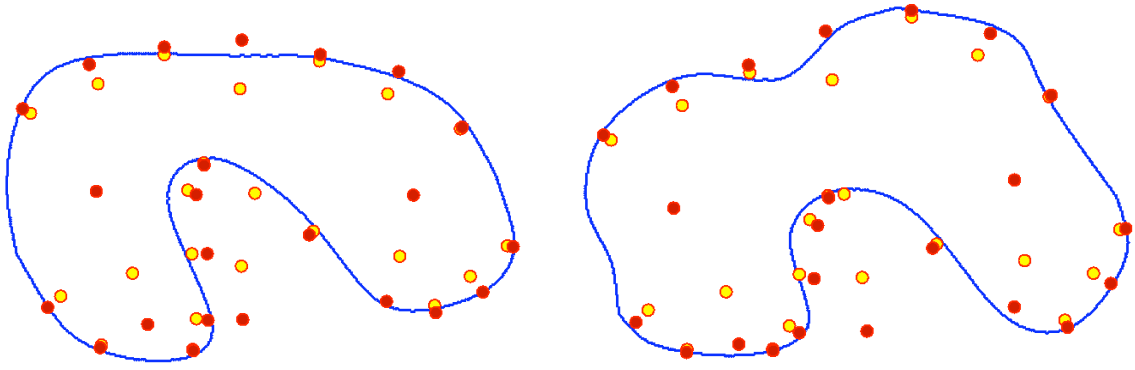


Figure 7.32: Virtual landmarks (red dots) and corresponding centroids (yellow dots) obtained for the training set of shapes in Fig. 7.12 for $\delta = 10$ and $a = 1$.

landmarks lie on their corresponding principal axis. The result is shown in Fig. 7.33 only with the virtual landmarks obtained. In this figure, we can notice that the distribution of landmarks among shapes is very similar, which indicates that landmarks are identified

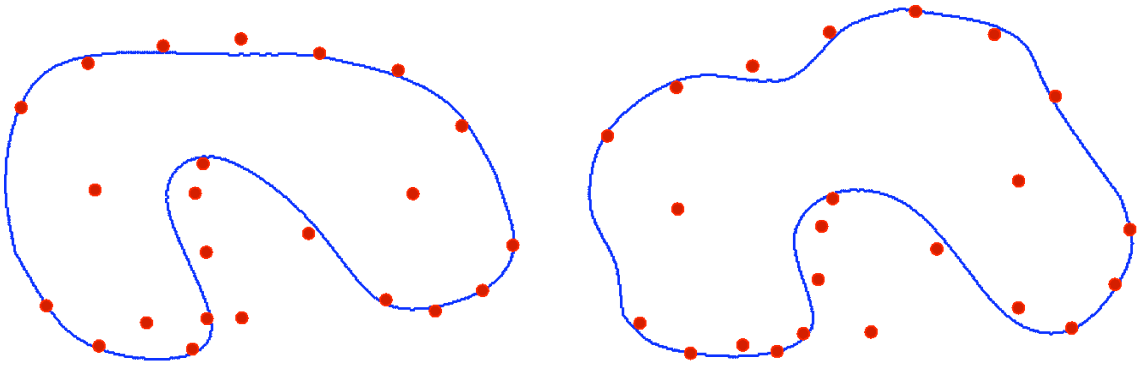


Figure 7.33: Virtual landmarks obtained for the training set of shapes in Fig. 7.12 for $\delta = 10$ and $a = 1$.

homologously in both shapes. If we use $\delta = 10$ and $a = 1.5$, most of the landmarks obtained seem to be located closer to the boundary, as can be seen in Fig. 7.34. In this figure, we have represented the centroids as well, to show that the distance from the centroids to the landmarks is greater than in the case shown in Fig. 7.32, since the parameter a is higher but the eigenvalues associated with each segment are the same.

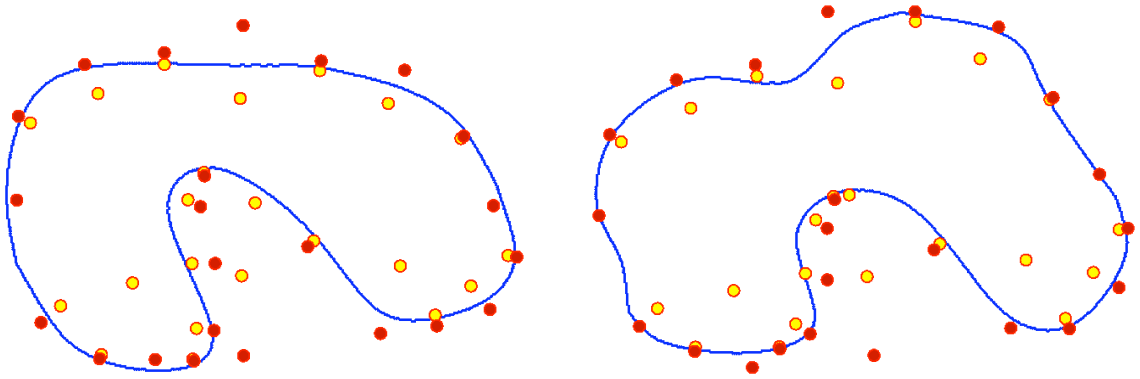


Figure 7.34: Virtual landmarks (red dots) and corresponding centroids (yellow dots) obtained for the training set of shapes in Fig. 7.12 with $\delta = 10$ and $a = 1.5$.

The distribution of landmarks for the case $\delta = 10$ and $a = 1.5$ is reproduced in Fig. 7.35 without the centroids. As in the previous case, the pattern obtained among shapes seems very similar, which highlights the correspondence of landmarks achieved for this approach.

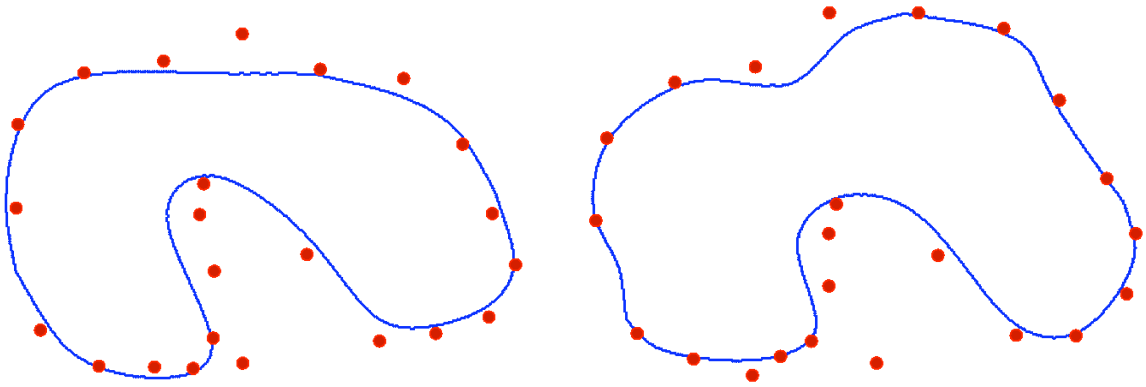


Figure 7.35: Virtual landmarks obtained for the training set of shapes in Fig. 7.12 for $\delta = 10$ and $a = 1.5$.

7.5 Hierarchical Nature of the RBS Method

There is a hierarchical component to the process of RBS that allows us to control the level of detail from a global to local manner. In the early stages of the method, landmarks are selected at a coarser level, whereas towards the end, the level of detail is finer. The final level of detail reached depends on the parameter δ considered. Higher values of delta retain only coarse features, whereas a small δ retains a higher level of detail among shapes in the training set. This is an important and strong feature of this family of methods. The hierarchical property can also be observed in the construction of the tree, as presented in Section 7.2. The tree has a rich global-to-local information that captures coarser features at the top and local features at the bottom. In the following, we illustrate this concept by giving three examples incorporating different levels of detail. In Figs. 7.36-7.38 we have represented the landmarks obtained with the PCA-based RBS approach for $\delta = 2.5$, $\delta = 10$, and $\delta = 25$, respectively.

We notice that the higher the parameter δ is, the lesser details are captured by the method. Furthermore, another feature of this family of methods is that the locations of the landmarks appearing at different levels of detail do not vary depending on the level of detail selected, because of this hierarchical property of the method. Only new landmarks are added if the level of detail increases, but the position of the landmarks previously captured at a

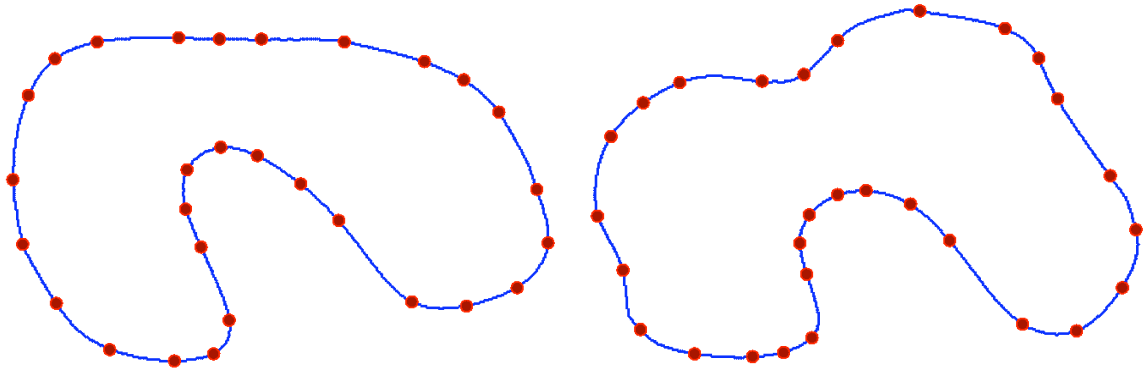


Figure 7.36: RBS based on PCA with $\delta = 2.5$ for the training set of shapes in Fig. 7.12.

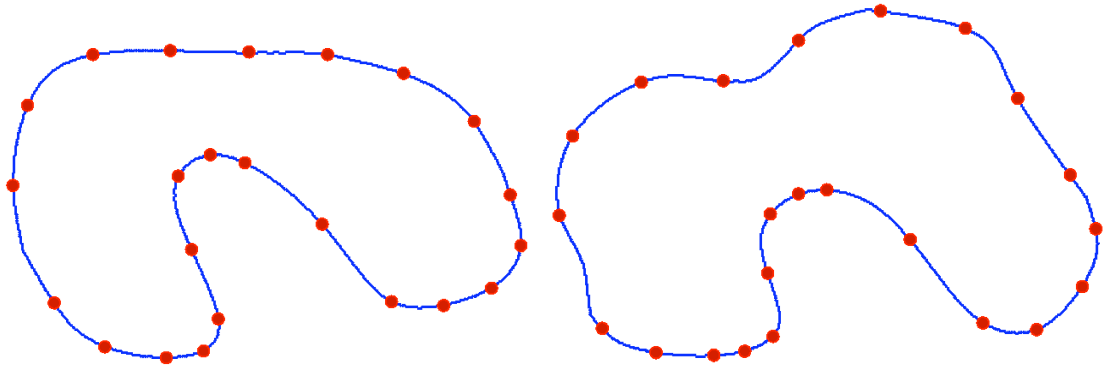


Figure 7.37: RBS based on PCA with $\delta = 10$ for the training set of shapes in Fig. 7.12.

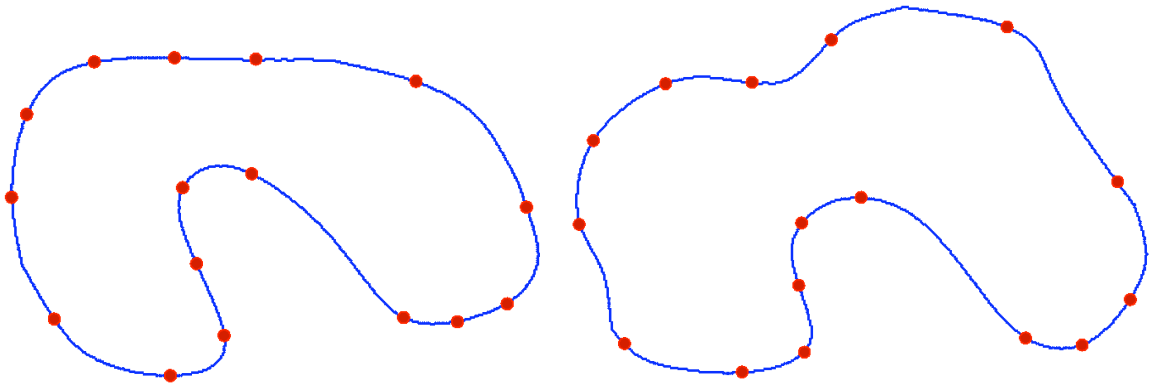


Figure 7.38: RBS based on PCA with $\delta = 25$ for the training set of shapes in Fig. 7.12.

coarser level remains invariant.

7.6 Conclusion

We have presented a family of methods based on global-to-local recursive boundary subdivision for determining landmarks on digital shapes. The method is general in being nD and in allowing the incorporation of a variety of shape analysis methods on the sub-segments to determine landmarks. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at higher resolution. This family of methods will be evaluated in Chapter 8.

Interestingly, the method when applied to any single shape will extract the main shape features and can be seen as a shape descriptor. No other shape will interfere as in the tagging process, and therefore, all the features captured will characterize that shape only. Note that the method naturally selects more landmarks in regions of higher curvature and fewer points in flat regions.

8

Evaluation of 2D Landmark Tagging Methods

In this chapter, we present the results of quantitative and qualitative evaluation of the automatic landmark tagging methods described in this part of the thesis. We introduced overall three novel families of methods to build 2D shape models from a training set of shapes; these are c-scale based (Chapter 5), variance based (Chapter 6), and recursive boundary subdivision based (Chapter 7). The c-scale based methods extract dominant points on the mean shape of the training set by using the concept of local curvature (Chapter 3). These landmarks are then propagated to all the shapes of the training set to establish correspondence in a local-to-global manner. The variance based method is guided by the strategy of equalization of the shape variance contained in the training set for selecting landmarks. This assures that more landmarks will be located where the variance in the training set of shapes is high, and less where the variance is low. The recursive boundary subdivision strategy is able to select dominant points according to the characteristics present among shapes of the training set. While being fundamentally different from the c-scale based and the variance based approaches, it incorporates both concepts by extracting dominant points while taking into account the existing variability of the training set of shapes.

The methods are applied to two training sets of shapes constituting two medical objects - the talus bone of the foot extracted from MR images, and the liver obtained after segmentation of abdominal CT volumes. The qualitative evaluation shows how each of the methods

achieves landmark selection on these two objects of interest. The quantitative analysis focuses on the compactness of a point distribution model for each training set of shapes. We want the models to be as compact as possible, and represent the existing variability of the training set of shapes with as few variation modes and landmarks as possible. Several new compactness factors are introduced to assess the performance of the methods independently of the number of modes and the number of landmarks considered. We compare the methods among manual and equally spaced annotations, which are the approaches commonly used for comparison purposes, the manual annotation being considered the ground truth.

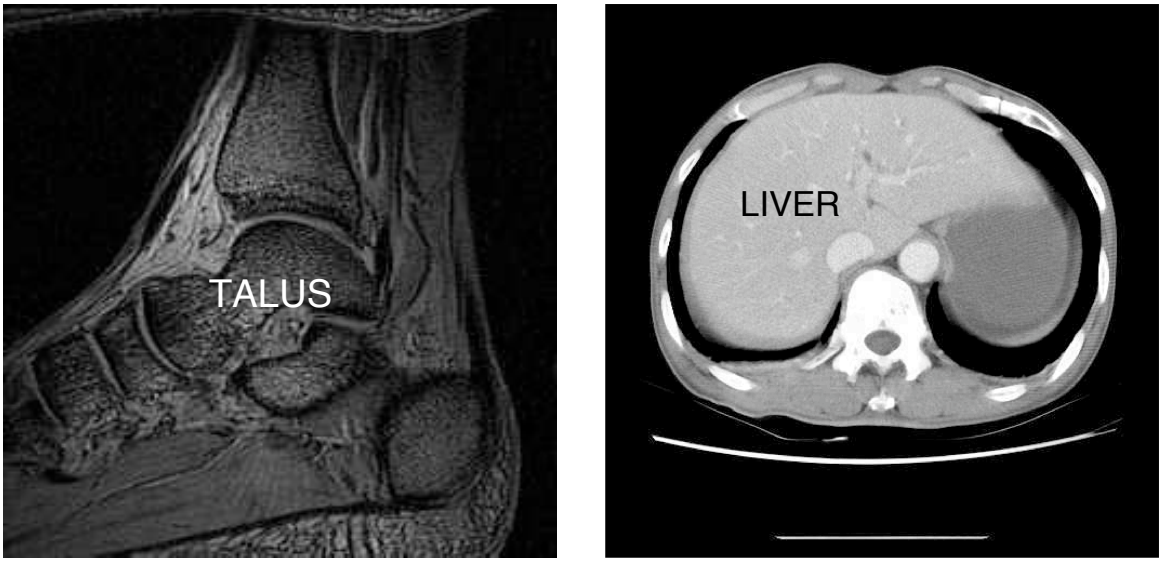
We first describe the image data sets that are used in evaluation. The metrics of model efficacy are presented in Section 8.2. Section 8.3 presents other methods used for comparison. The qualitative and quantitative comparisons are performed in Sections 8.4 and 8.5, respectively. The conclusions are stated in Section 8.6.

8.1 Description of Image Data Sets

We tested the proposed methods on two data sets derived from medical CT and MR imaging. The first data set consisted of 40 MRI foot images, the object of interest being the talus bone of the foot (Fig.8.1(a)). The images correspond to approximately the same anatomic plane in the foot among different subjects. The second data set was similarly formed from 40 CT liver images (Fig.8.1(b)) also selected from equivalent anatomic planes among different subjects. All the objects of interest were segmented by using the operator-steered Live Wire [Falcão et al., 2000] technique to create the two training sets of talus and liver shapes.

8.2 Metrics of Model Efficacy

The compactness factor employed here is such that a model is considered to be more compact if it can capture the variations existing in the training shapes with as few modes and landmarks as possible. This idea can be captured with the following formulation.



(a) (b)

Figure 8.1: The objects of interest. (a) The talus bone of the foot in an MR image. (b) The liver in an abdominal CT image.

The manner in which a model captures the variations present in the training shapes is a function of the number l of modes selected from all the modes observed in the principal component analysis of the point distribution model [Cootes et al., 1995]. Fundamentally, it also depends on the number n of landmarks selected. Therefore, we express the compactness of a model in terms of both these variables as a fraction of the variation per mode to total variation, given by

$$\chi_{n,l} = \frac{\sum_{i=1}^{l} \lambda_i}{\frac{2n}{\sum_{i=1} \lambda_i}}, \tag{8.1}$$

where λ_i is the eigenvalue representing the i^{th} largest mode. As per principal component analysis, the larger an eigenvalue, the greater is the variation in the mode associated with that eigenvalue. $\chi_{n,l}$ indicates how well the variation is captured as a function of *both* the number of eigenvalues (modes) and the number of landmarks selected. $\chi_{n,l}$ is shown as a method of modelling surface plot in Fig.8.2 for the hierarchical selection and parametric propagation in the talus bone training set based on c-scale principles.

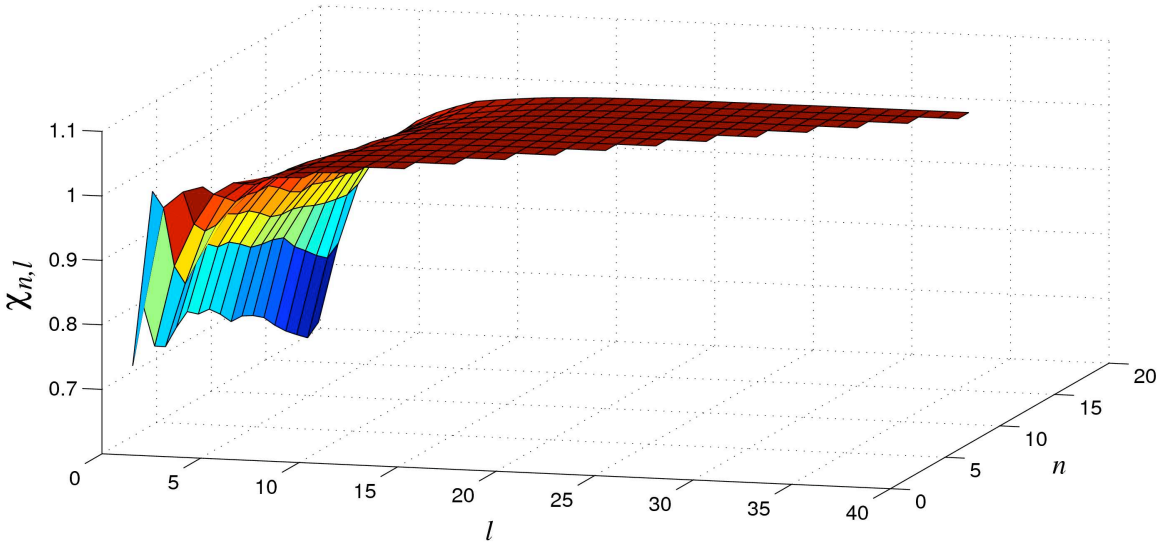


Figure 8.2: $\chi_{n,l}$ for $n = 1, ..., 28$ and $l = 1, ..., 56$. In this case, only valleys were considered for the c-scale based landmark tagging approach. These landmarks were selected on the mean shape of the talus bone in a hierarchical manner and propagated to the individual shapes in the talus data set by using the parametric method.

It will be difficult to compare methods by using only $\chi_{n,l}$, since, from the representation using surfaces, it becomes difficult to clearly discern the differences among methods. To carry out the comparison independently of the number of modes, we calculate the area under the curve for each of the cases $n = 1, ..., L$, where L is the maximum number of landmarks considered in the study. If we integrate $\chi_{n,l}$ over the l variable, we obtain

$$\chi_n = \frac{\sum_{l=1}^{2n} \chi_{n,l}}{2n}. \tag{8.2}$$

If we integrate χ_n over L landmarks, then we can express the overall ability of the method to capture variation by using up to L landmarks by:

$$\chi = \frac{\sum_{i=1}^{L} \chi_i}{L}. \tag{8.3}$$

In the remainder of this section, we will use equations (8.2) and (8.3) to compare among all the methods studied for both objects. Only sample comparative results will be presented and not an exhaustive set of all the results of all evaluations.

8.3 Comparison With Other Methods

We decided to compare the proposed methods to manual and equally spaced annotations mainly because of the complexity and difficulty of reproducing other techniques such as MDL and dense point correspondence strategies. Further, the former are the most commonly used as references in the literature. In this section, we will describe how manual and equally spaced annotations are obtained and what their main issues are.

8.3.1 Manual Landmark Tagging

The manual method is implemented as follows:

1. Print all training images on paper.
2. Examine all shapes to get a qualitative understanding of the homologous points that may be selected on all of them.
3. Mark anatomically the same, most prominent appearing points, as many as we can, on the boundary of each training shape.
4. Select the same points, as closely as possible as those marked on paper, on the images in the computer.
5. Develop a table of the landmark points corresponding to the different number of landmarks considered in our experiments. This is done to avoid repeating steps 1-3 when we vary the number of landmarks.

The process needs to be repeated every time we increase the number of landmarks, since the selection of landmarks is based on the most prominent features among the shapes of the

training set. We denote prominent points in a shape as those landmarks that can be easily identified, such as anatomical landmarks that have a certain meaning for the shape, high curvature regions, corners, or any other features that can help in finding corresponding landmarks among shapes. Sometimes, other landmarks need to be added in between prominent landmarks to be able to describe all shapes consistently.

It is worth stressing that not all the shapes of a given family present the same number of features. Therefore, manual annotation is a highly subjective task, and the results depend on the expert's attitude and how well he/she identifies landmarks and establishes correspondences among shapes. This is a highly intuitive and difficult task, as one needs to consider all shapes in the training set simultaneously, and annotate them all the same way to retain landmark correspondence among shapes consistently. (In 3D, manual annotation is impossible to achieve for these reasons.)

Following the above systematic procedure is essential in order to do full justice to the manual method. The results obtained from manual annotations are usually considered as ground truth to evaluate other automatic methods. The idea is for the manual annotation to serve as a reference to measure how other methods perform at an intuitive level.

8.3.2 Equally Spaced Landmark Tagging

The equally spaced method has been implemented as follows:

1. Take the first point selected in the manual method as the initial point.
2. Given the number of landmarks, starting from the initial point, space the points as equally as possible along the boundaries, coming back to the initial point.

By considering different number of landmarks, the spacing between consecutive landmarks varies. Therefore, the process needs to be repeated if we vary the number of landmarks, as their position on the shapes will change. Furthermore, this method is highly dependent on the

starting point considered, which needs to be selected manually on all the shapes, respecting landmark correspondence among shapes.

8.4 Qualitative Comparison

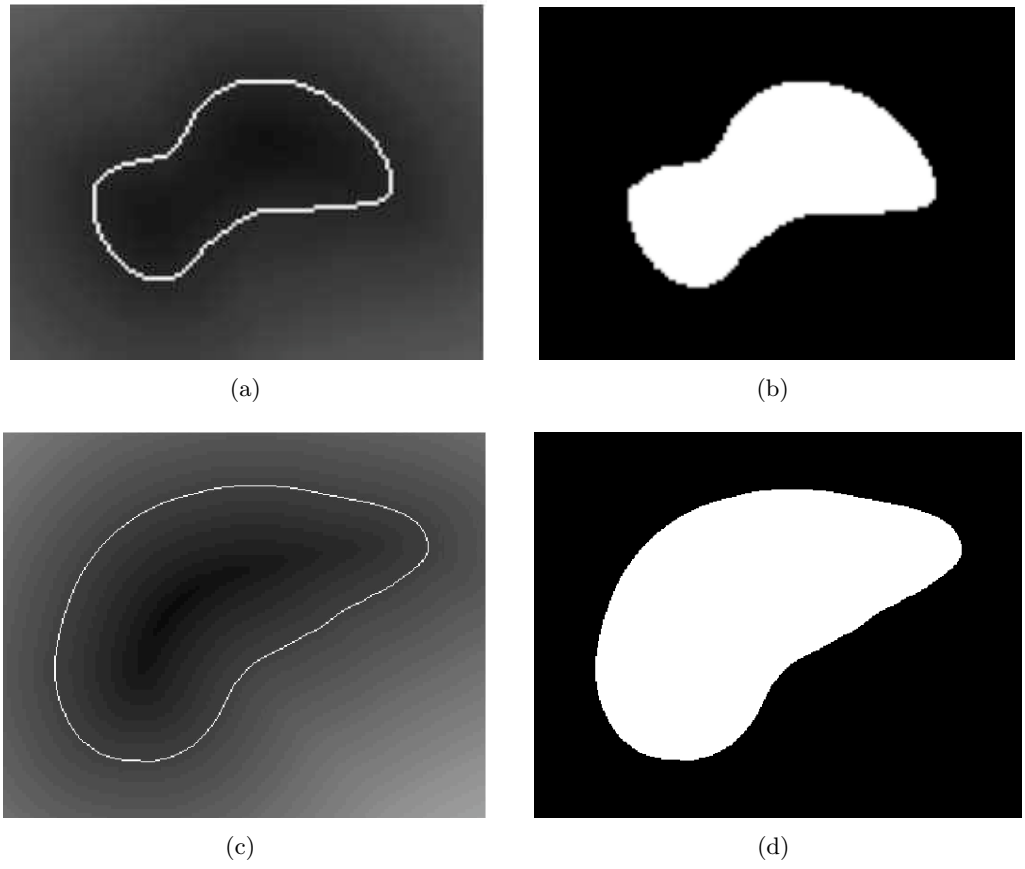
In this section, we show how the landmarks are localized by the the three groups of methods presented in Chapters 5-7. The groups are very different and the landmarks resulting from each group do not necessarily have the same meaning. In the curvature scale-based approach (Chapter 5), landmarks are features of the shape, selected directly on the mean shape and based on curvature. In the variance equalization strategy (Chapter 6), landmarks are distributed around each shape, placing more landmarks in regions where the variance among shapes is high and less where it is low, taking into account the real variability existing in the training set. In this case, landmarks are not features of the shape but of the variability existing in the training set of shapes. The recursive boundary subdivision family incorporates a variety of shape analysis methods on the sub-segments to localize landmarks. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at higher resolution. More landmarks will be selected at locations where more features exist among the shapes of the training set. Therefore, this last method considers landmarks as shape features, but will consider more or less number of landmarks taking into account the variability that exists among shapes of the training set more globally initially and subsequently honing in on local segments.

8.4.1 Curvature Scale-Based Landmark Tagging

This approach works in two distinct steps. First, the mean shape of the training set is obtained, and then, the landmarks are selected directly on the boundary of the mean shape of the training set by using the c-scale shape descriptor presented in Chapter 3. To generate a PDM, we propagate these landmarks to all the shapes of the training set. That way we assure that all the shapes have the same number of landmarks n and in "corresponding" locations.

Mean Shape Extraction

The mean shape of each of the training sets was estimated from the 40 training binary images and computed by using Euclidean signed distance transform operations based on the Euclidean Distance Transform (EDT) by [Breu et al., 1995], as explained in Section 5.3. Fig. 8.3.(a) shows the averaged distance image for the talus bone and the boundary of the mean shape superimposed as a white contour. Fig. 8.3.(b) shows the mean shape mask for the talus obtained after thresholding the distance image at 0. The results for the liver are shown in Figs. 8.3.(c)-(d).



(a) (b)

(c) (d)

Figure 8.3: Constructing the mean shape from a training set of images: (a-b) the talus bone obtained from 40 segmented MRI images of the foot; (c-d) the liver obtained from 40 segmented CT images. (a),(c) Mean distance image with representation of the mean shape boundary (in white) corresponding to zero distance values. Inside this boundary, we have negative values and outside positive values. (b),(d) Mean shape obtained after thresholding the respective distance image at 0.

Landmark Selection

Different number of landmarks were selected on the mean shape for each of the training sets by using the c-scale method as described in Section 5.1. We demonstrate in Figs. 8.4 and 8.5 that it is easy to vary the number and type of landmarks selected, which is a characteristic of the c-scale approach and which can be very useful in different applications. Figs. 8.4 and 8.5 show different number of landmarks selected on the mean shape of the talus bone and of the liver, respectively.

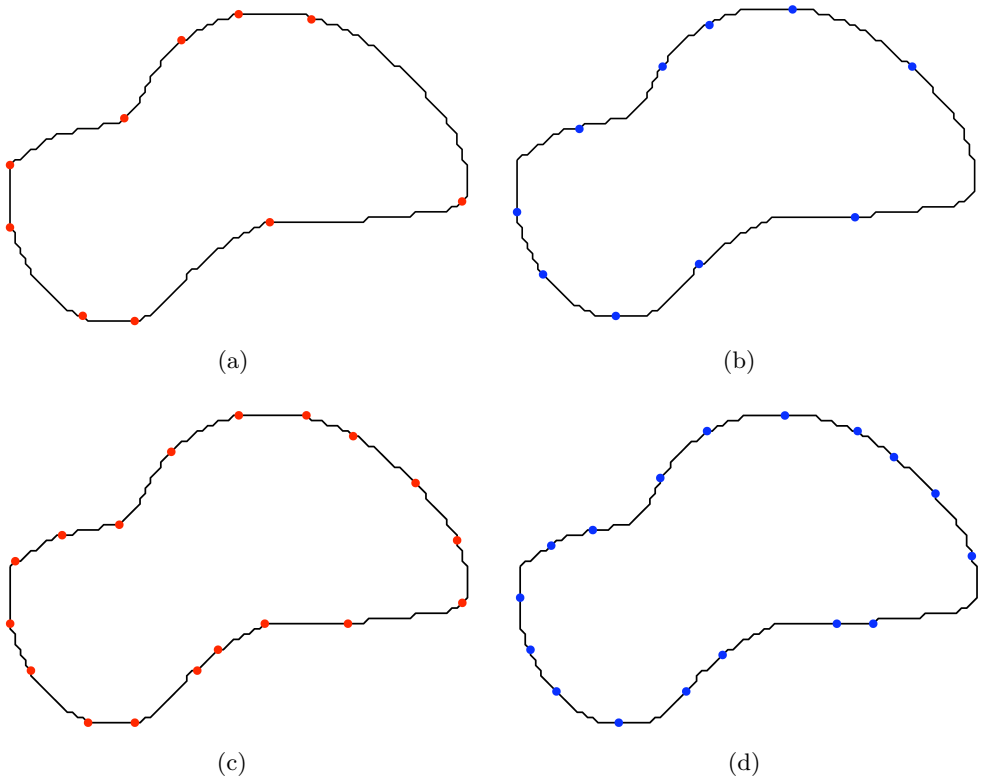


Figure 8.4: Landmark selection via the c-scale approach on the mean shape of the talus bone. (a) 10 landmarks selected considering only valleys ($t = 2$, $w = 2$, $m = 1$, $se_v = 5$, and $\theta_v = 0$). (b) 10 landmarks selected considering only peaks ($t = 2$, $w = 2$, $m = 1$, $se_p = 5$, and $\theta_p = 0$). (c) 18 landmarks selected considering only valleys ($t = 1$, $w = 2$, $m = 1$, $se_v = 5$, and $\theta_v = 0$). (d) 17 landmarks selected considering only peaks ($t = 1$, $w = 2$, $m = 1$, $se_p = 5$, and $\theta_p = 0$).

Figs. 8.4(a) and 8.4(c) show the selection of high curvature points (valleys) in the talus bone of the foot, for 10 and 18 landmarks selected, respectively. Similarly, Figs. 8.4(b) and

8.4(d) present the results of selecting inflection points (peaks), for 10 and 17 landmarks, respectively. As we can observe from the parameters considered, the only parameter that changed to obtain more or less number of landmarks was the scale t. All the other parameters were fixed for that particular application. For smaller values of t we obtain a higher number of dominant points.

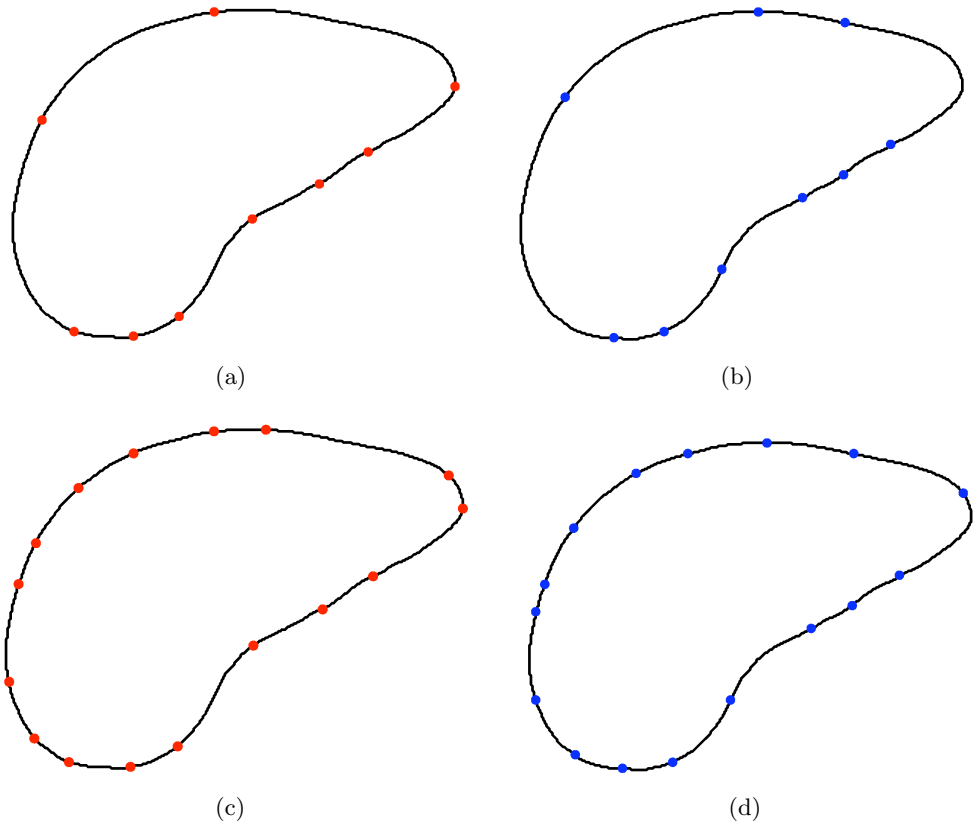


Figure 8.5: Landmark selection via the c-scale approach on the mean shape of the liver for $t = 2.5$, $w = 3$, $m = 2$, and $se_v = se_p = 5$. (a) 9 landmarks selected considering only valleys ($\theta_v = 1$). (b) 9 landmarks selected considering only peaks ($\theta_p = 1.3$). (c) 16 landmarks selected considering only valleys ($\theta_v = 0$). (d) 16 landmarks selected considering only peaks ($\theta_p = 0$).

Similarly, in Fig. 8.5, we can observe the selection of different number of landmarks on the mean shape of the liver training set. In this case, Figs. 8.5(a) and 8.5(c) show the selection of high curvature points (valleys) on the mean shape, selecting 9 and 16 landmarks, respectively. Similarly, Figs. 8.5(b) and 8.5(d) represent the inflection points (peaks) for 9

and 16 landmarks, respectively. In this case, the selection of different number of landmarks is achieved by varying θ_v and θ_p, to retain only the most prominent peaks or valleys for that parameter selection. All the other parameters were fixed. As we can observe, the landmarks obtained at a lower resolution, are also represented at a higher resolution at exactly the same position, since we are just retaining the most prominent peaks and valleys that were captured for a certain parameter configuration. This shows another way of varying the number of landmarks selected, as explained in Sections 3.3.2 and 5.4.

From these images we can see that the method can capture different features of the (mean) shape and with different levels of detail, allowing the selection of any desired number of landmarks required for the application. The selection of landmarks can be based on high curvature points (valleys) only, inflection points (peaks) only, or both simultaneously depending on the application.

8.4.2 Landmark Tagging Via Variance Equalization

The first step of the approach consists of calculating the variance after parameterizing the shapes of the training set, as explained in Section 6.2. For the talus bone, the total variance among shapes is presented in Fig. 8.6, after reparameterization of 40 training shapes.

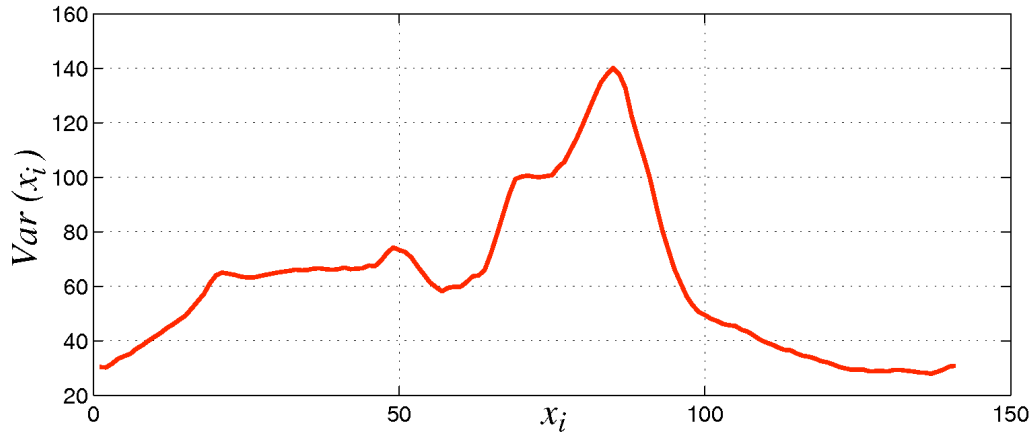


Figure 8.6: Variance among the parameterized training shapes of the talus bone.

To see where the highest variance occurs among shapes, we display in Fig. 8.7 two talus shapes where we have located x_i for $i = 1, 50$, and 100.

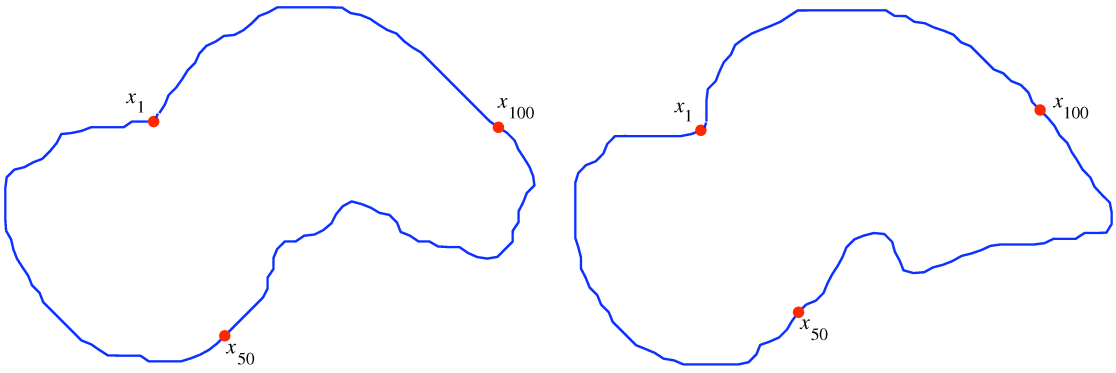


Figure 8.7: Two shapes of the talus bone with corresponding points for different x_i. Compare locations of x_i in Fig. 8.6.

We may observe that the highest variance occurs between x_{50} and x_{100}, the bottom part of the shapes, where more landmarks should be tagged. Some of the shapes in the set are displayed in Fig. 8.8, overlaid, to show the variability as well as the correspondence among landmarks. In the upper part of the structures, where variance is much lower, the points

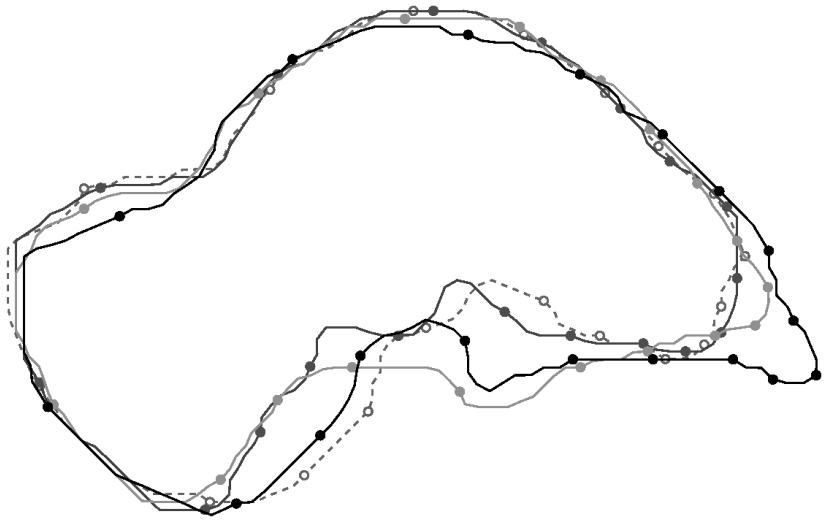


Figure 8.8: 18 landmarks detected on several training shapes of the talus bone via variance equalization.

tagged are farther apart than in the lower part.

The results of variance equalization for two different shapes, considering different number of landmarks, are presented in Fig. 8.9. We observe that, independently of the number of landmarks considered, more landmarks are selected at the bottom part than at the upper part of the shapes.

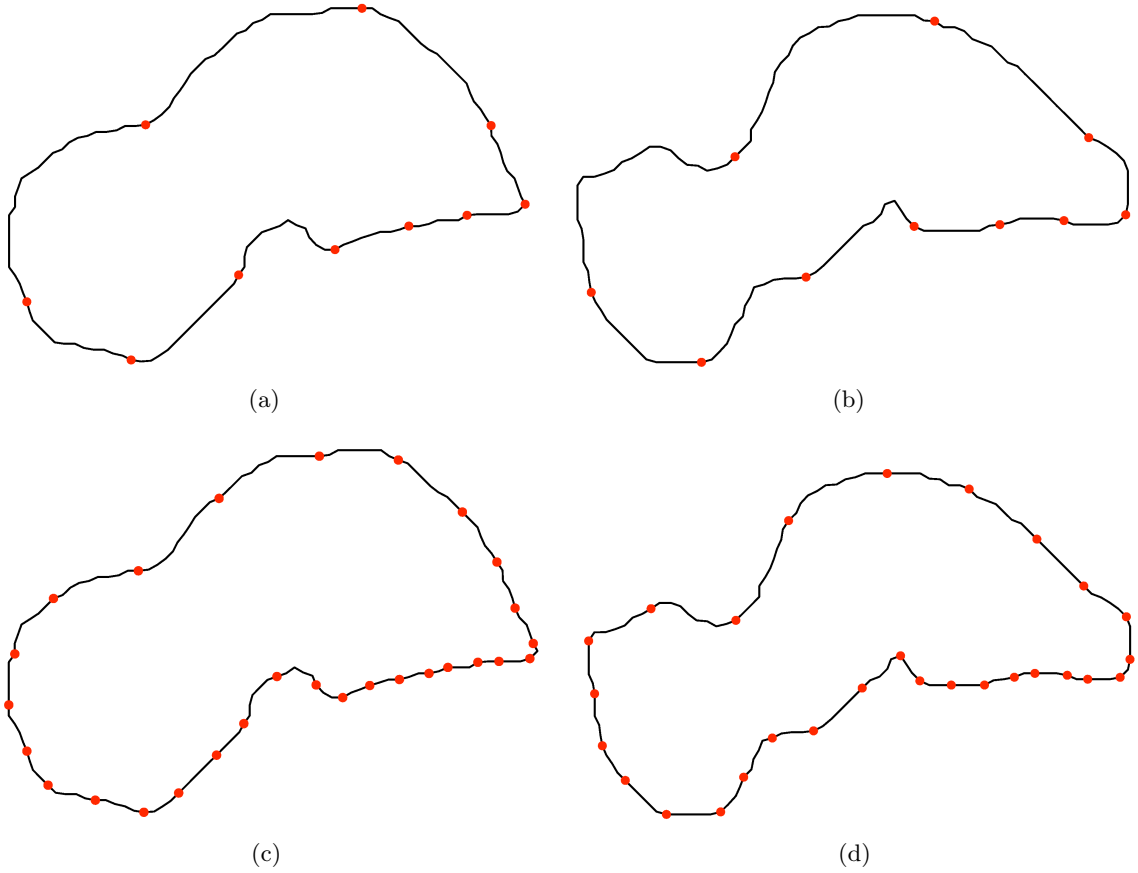


Figure 8.9: Landmark selection via variance equalization on two training shapes of the talus bone: (a-b) 10 landmarks, and (c-d) 28 landmarks.

We plot the equalized variance among training shapes, corresponding to the selected landmarks, in Fig. 8.10. Note that the variance, for all the landmarks considered, is effectively equalized among the shapes of the training set.

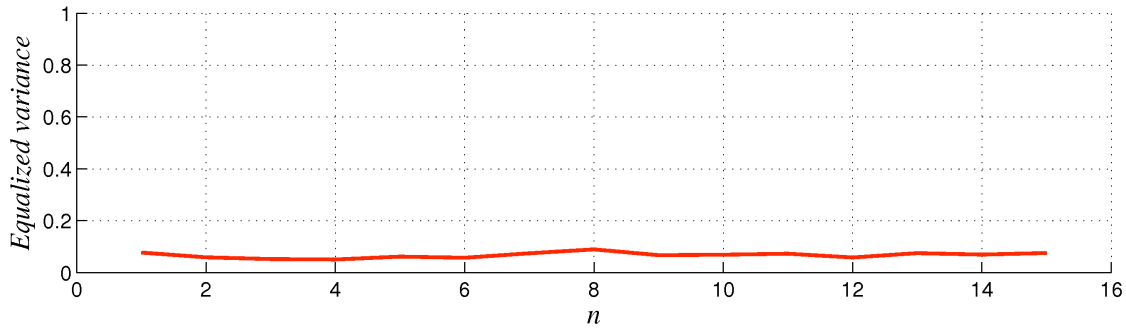


Figure 8.10: Equalized variance of the talus training shapes for 15 landmarks.

If we apply the same method to the liver, we obtain the variance plot of Fig. 8.11.

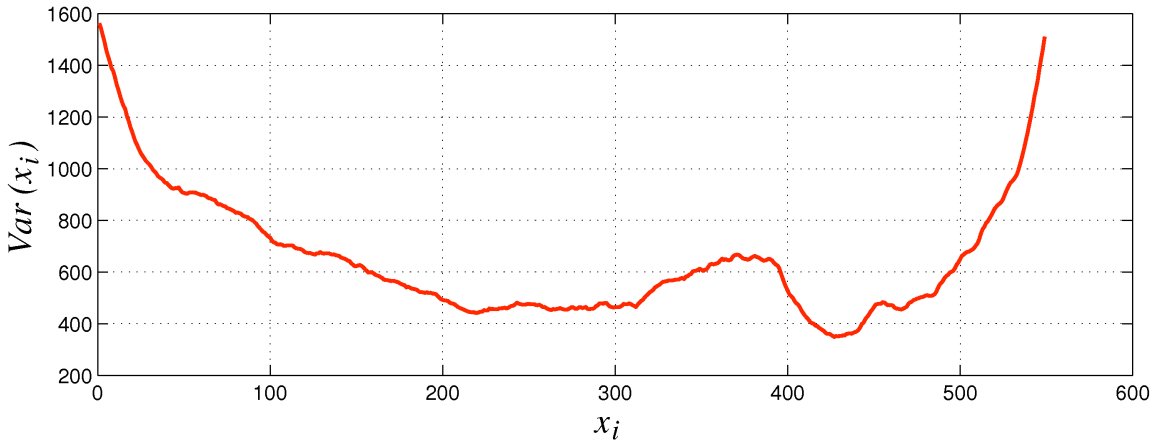
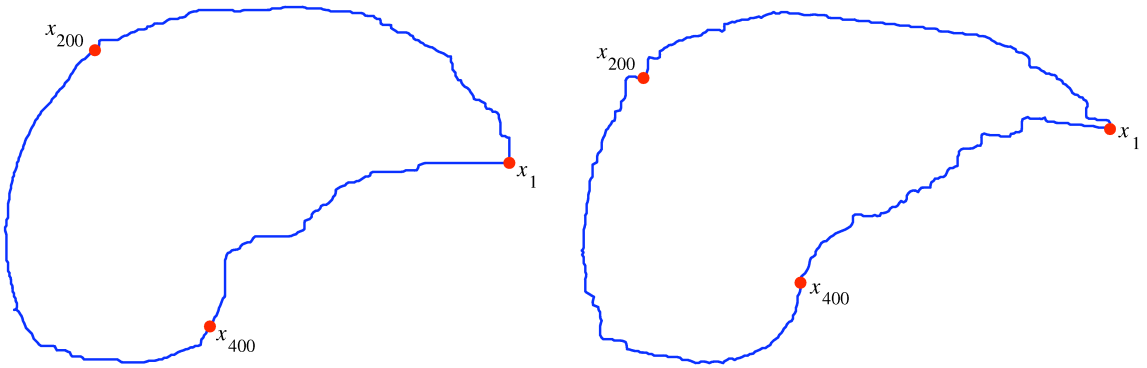


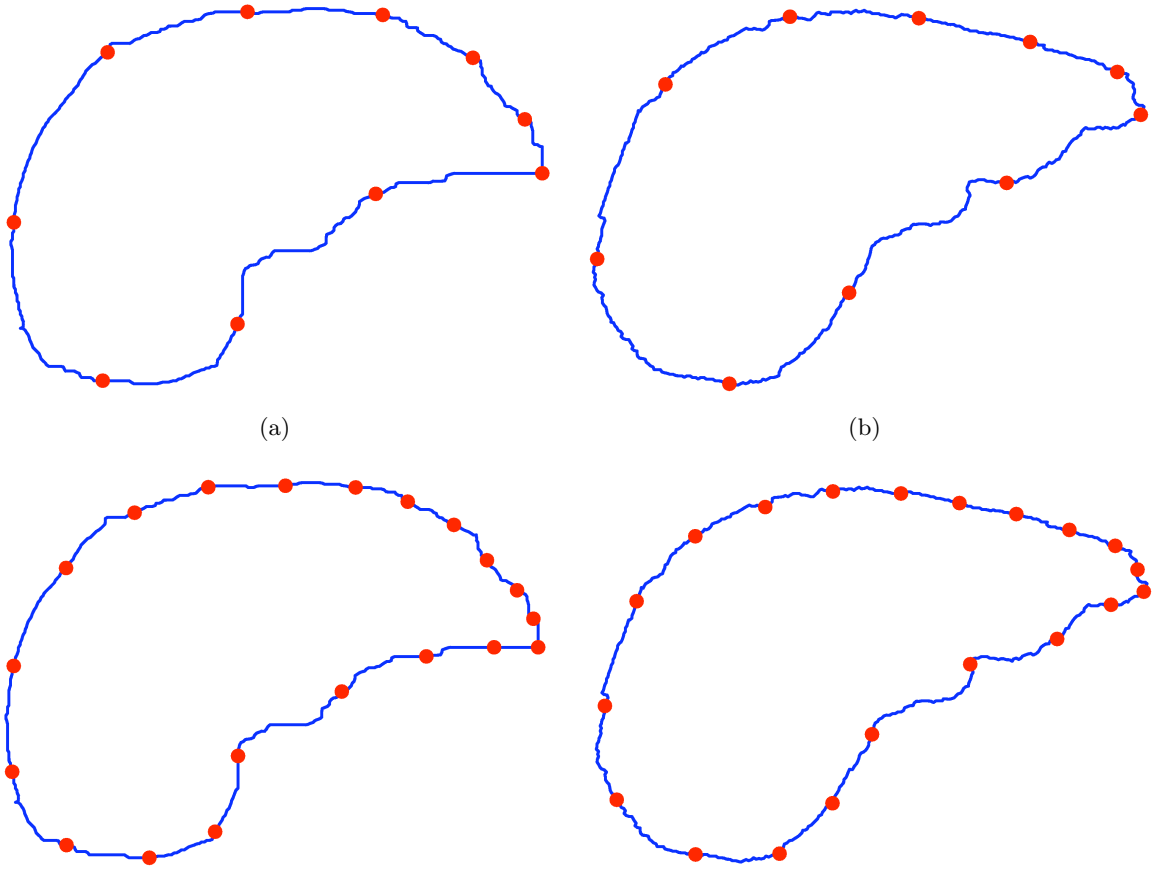
Figure 8.11: Variance among parameterized training shapes of the liver.

To see where the highest variance occurs among shapes, we present in Fig. 8.12 two liver shapes where we have located x_i for $i = 1, 200$, and 400. In this case, the highest variance occurs around x_1, which means that the most variable part in the training set of shapes is around the right hand side of the liver shapes. This can also be seen in Fig. 8.13, where more landmarks are located in that region and less in the rest of the shape, independently of the number of landmarks considered. The spacing between landmarks is smaller in high variance regions.



(a) (b)

Figure 8.12: Two liver shapes with corresponding points for different x_i. Compare locations of x_i in Fig. 8.11.



(a) (b)

(c) (d)

Figure 8.13: Landmark selection via variance equalization on two liver training shapes: (a-b) 10 landmarks, and (c-d) 20 landmarks.

As for the talus, the variance calculated by using the selected landmarks is represented in Fig. 8.14. We observe that the resulting curve is flat, which means that the variance is properly equalized by the landmarks selected.

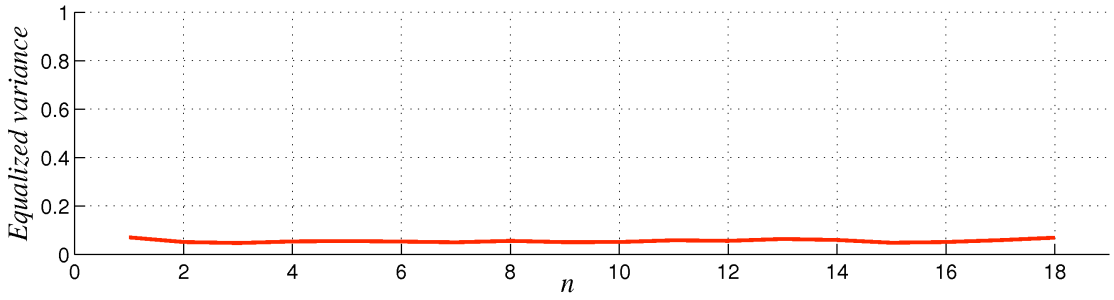


Figure 8.14: Equalized variance of the liver training shapes with 18 landmarks.

The results obtained using this strategy are independent of the initial point selected to parameterize the shapes of the training set, as the starting point of the landmark tagging method will always be at a point corresponding to the maximum of the variance plot. This value does not vary when we select different starting points and it is found automatically in each case.

8.4.3 Landmark Tagging Via Recursive Boundary Subdivision Approaches

The recursive boundary subdivision family (Chapter 7) incorporates a variety of shape analysis methods (by selecting α and β) on the sub-segments to determine landmarks, as presented in Section 7.3. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at higher resolution. In the following, we will study several variants of the method and see how they select landmarks on the two shape families of interest.

Distance-based RBS

In this variant of the RBS method presented in Section 7.3.1, α and β are based on distance. Method α selects the two farthest points on each training shape. Method β selects

the farthest distance from the line associated with the segment on each training shape. This distance has to be greater than a threshold δ, which is the stopping criterion for the method and decides which shape of the training set will lead the subdivision process. Once the shape is determined, that landmark is propagated to all other corresponding segments of the training set by using the closest point propagation strategy. In Fig. 8.15, we show the selection of landmarks obtained by using distance-based RBS on the training set of the talus bone.

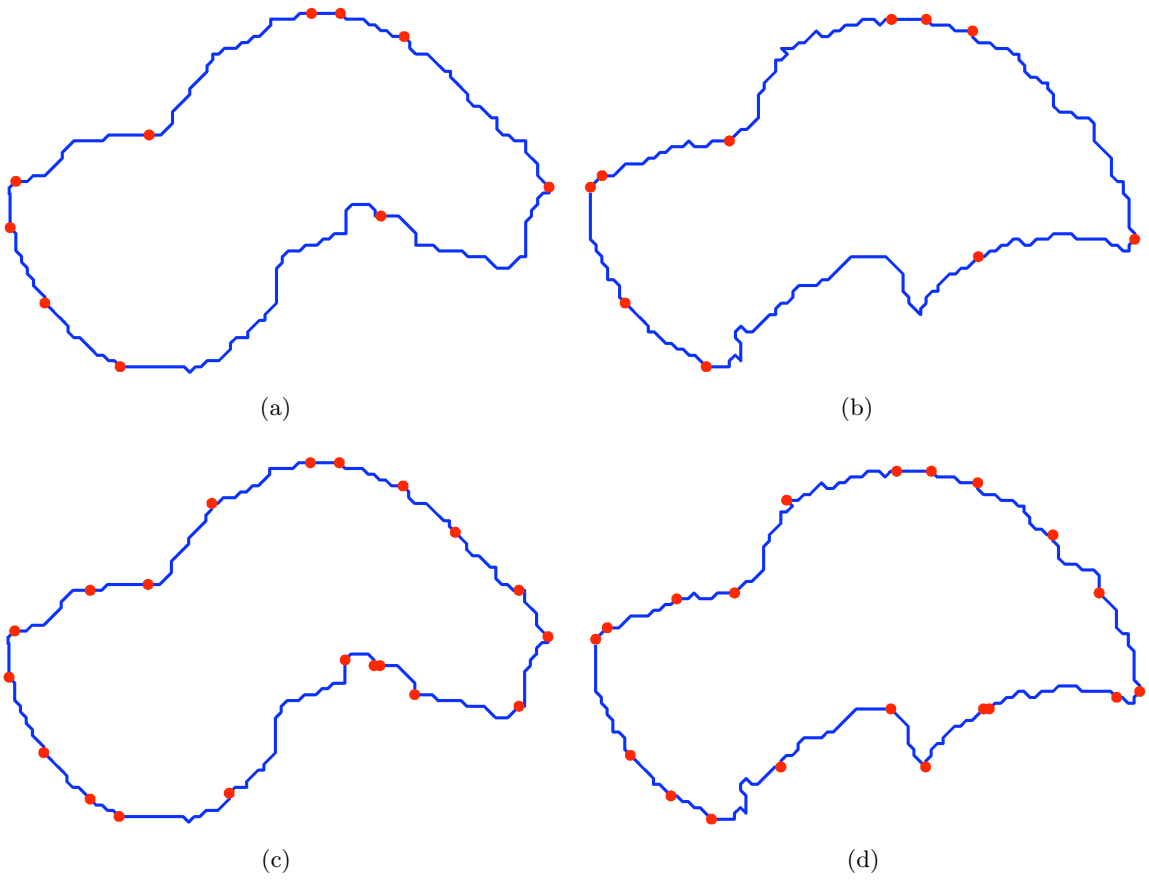


Figure 8.15: Landmark selection via Distance-based Recursive Boundary Subdivision on two training shapes of the talus: (a-b) 10 landmarks, and (c-d) 20 landmarks.

Similarly, Fig. 8.16 presents the selection of landmarks for the liver training set. For both talus and liver, we have varied the number of landmarks to show that for a higher number of landmarks, the method captures a higher level of detail. Furthermore, the method is multi-

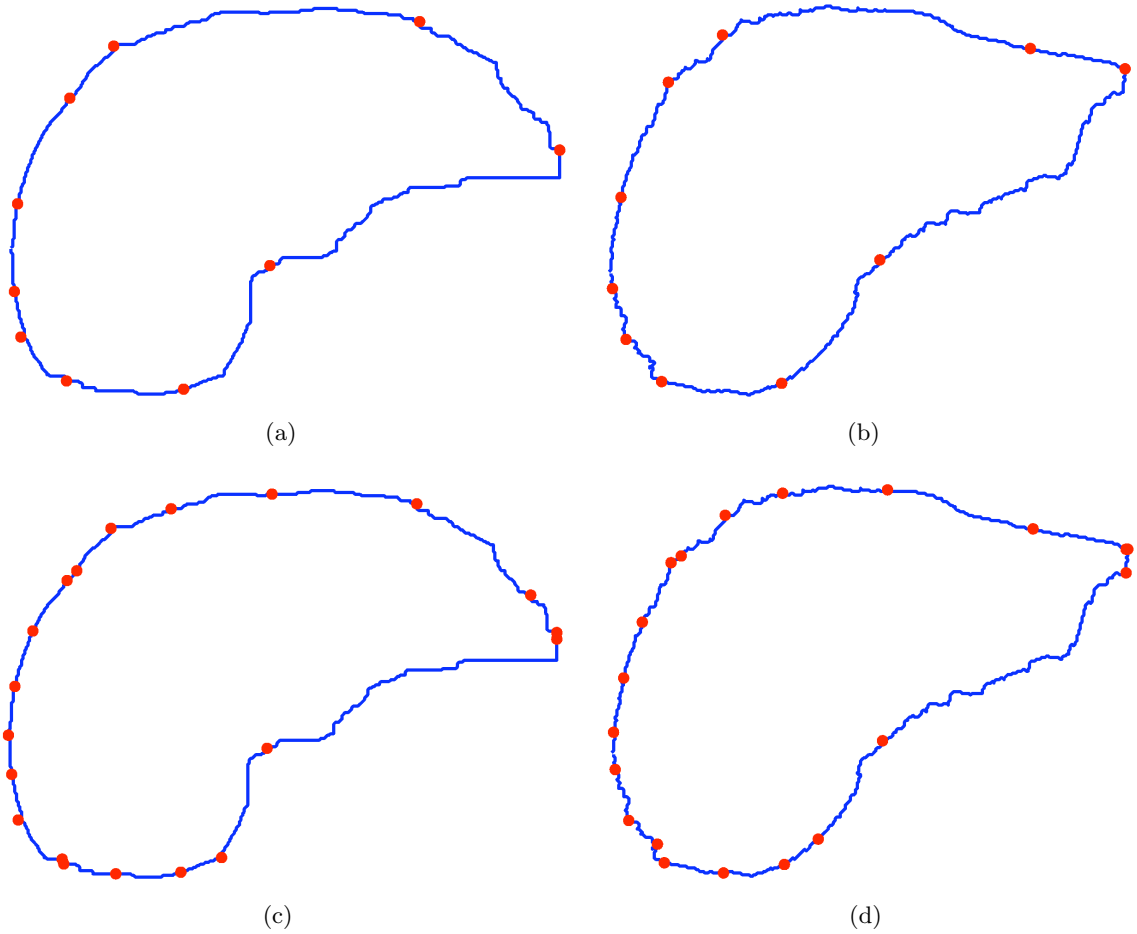


Figure 8.16: Landmark selection via Distance-based Recursive Boundary Subdivision on two liver shapes: (a-b) 10 landmarks, and (c-d) 20 landmarks.

resolutional and the landmarks obtained at a lower resolution are a subset of those selected at a higher resolution. In this case, we are showing only two of the 40 shapes of each training set, which means that some of the landmarks are introduced by features appearing on other shapes of the training set, and not necessarily on these two shapes only.

PCA-based RBS

As an alternative to the distance-based RBS strategy, we implemented the PCA-based RBS approach, as explained in Section 7.3.2. In this case, the two initial points determined by method α, are taken on each boundary, in the direction of the first principal (inertia) axis of

the shape boundary. Method β determines new landmarks on each corresponding segment in the direction of the second principal axis. In this case, the parameter δ is the eigenvalue associated with the second inertia axis of each segment, and is used to decide if subdivision should continue. Even if only one eigenvalue among all shapes is bigger than δ, the subdivision is applied to all corresponding segments.

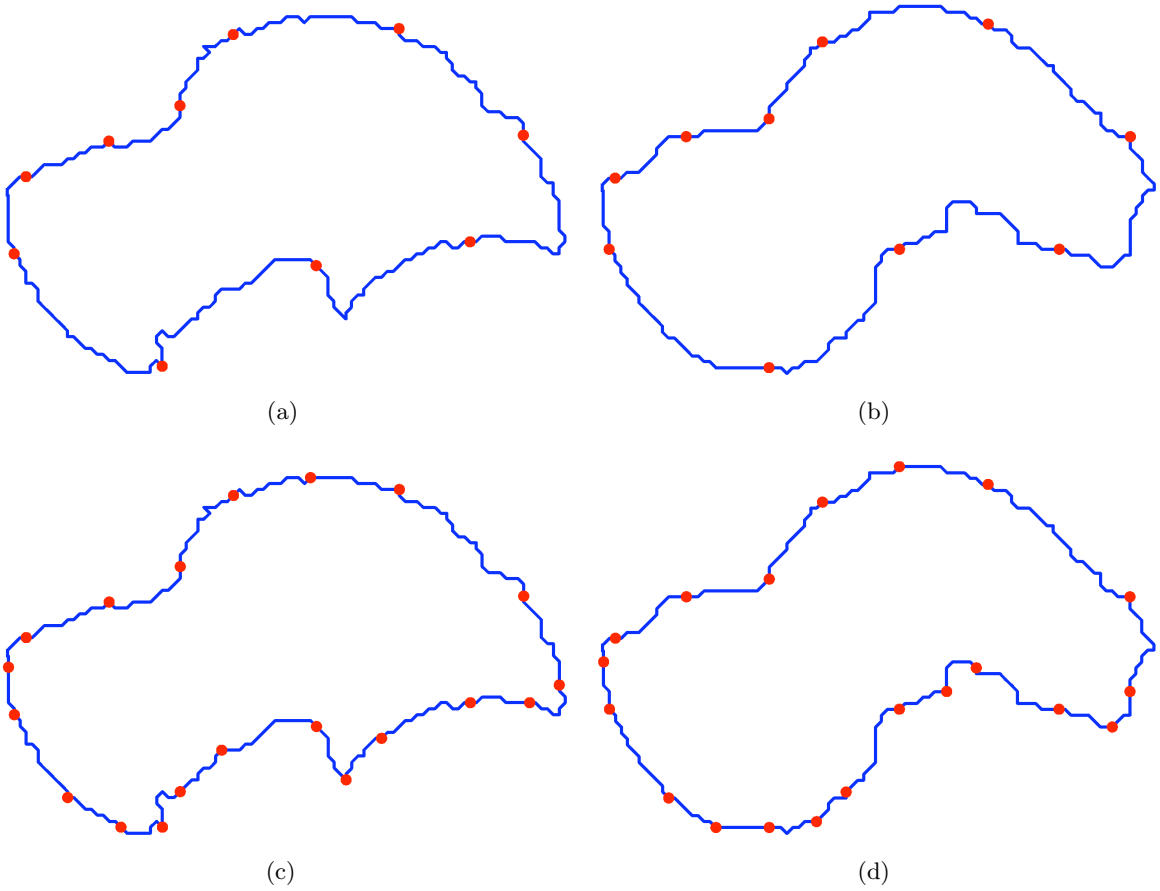


Figure 8.17: Landmark selection via PCA-based Recursive Boundary Subdivision on two training shapes of the talus bone: (a-b) 10 landmarks, and (c-d) 20 landmarks.

Results from this method, for 10 and 20 landmarks, are shown in Figs. 8.17 and 8.18 for the talus bone and the liver, respectively. We observe that the principal components seem to capture shape features at remarkably corresponding positions. The correspondence seems to obey our intuitive thinking for these shapes. As for the distance-based method,

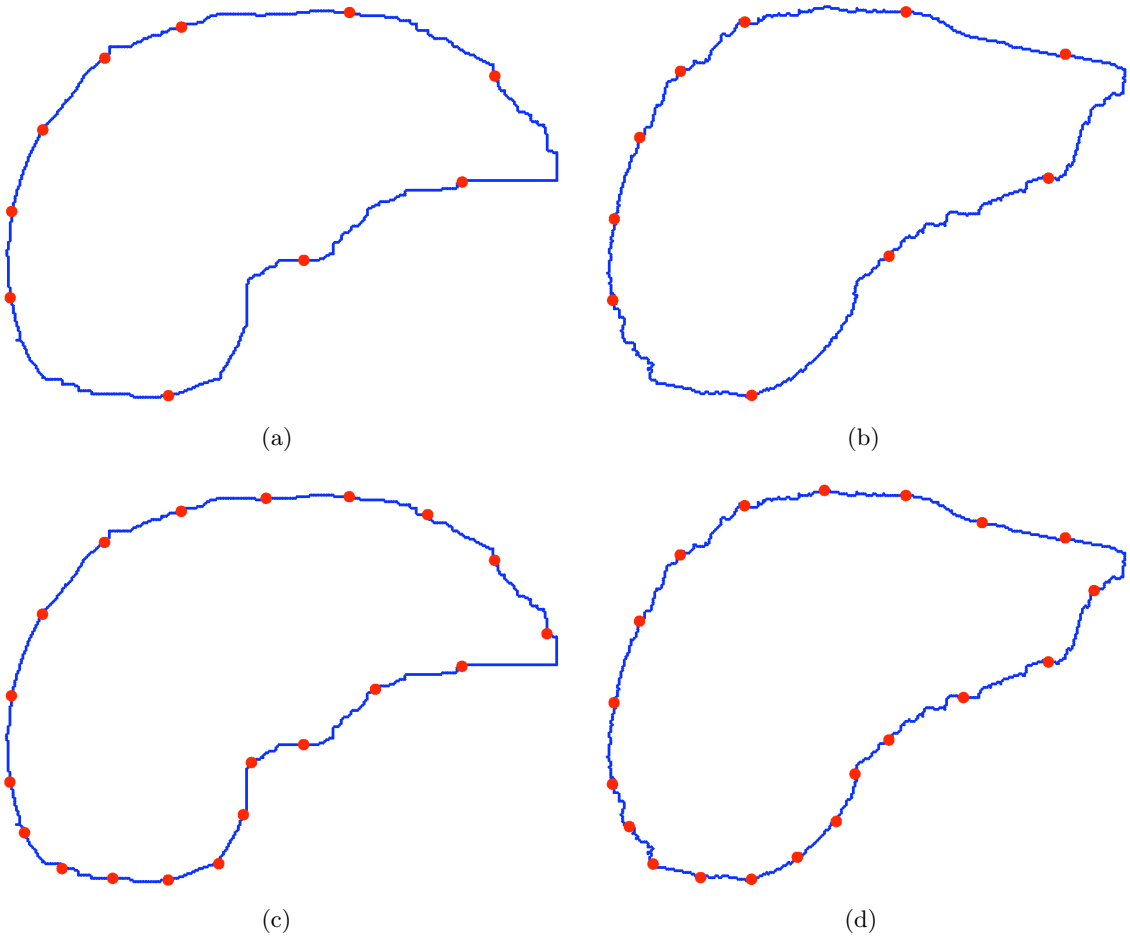


Figure 8.18: Landmark selection via PCA-based Recursive Boundary Subdivision on two training shapes of the liver: (a-b) 10 landmarks, and (c-d) 20 landmarks.

the landmarks obtained at a lower resolution are preserved at higher resolutions, where only more level of detail is incorporated into the shape description of each shape. This variant seems to perform better than the distance-based approach, and avoids the use of the closest-point method to find corresponding landmarks among shapes. The closest-point approach is dependent on how well the shapes are registered, whereas the PCA approach is more robust to registration errors and does not need to be applied on registered training sets to find corresponding landmarks.

Virtual RBS

Virtual RBS is the last variant of the RBS family, and it is described in Section 7.4. It uses the same strategy as the PCA-based RBS, but instead of confining the landmarks to the boundary, these are located at virtual positions with respect to it. The subdivision of the segments at each iteration follows the same principle as the PCA-based RBS. In particular, we present two different approaches to obtain virtual landmarks. The first variant considers

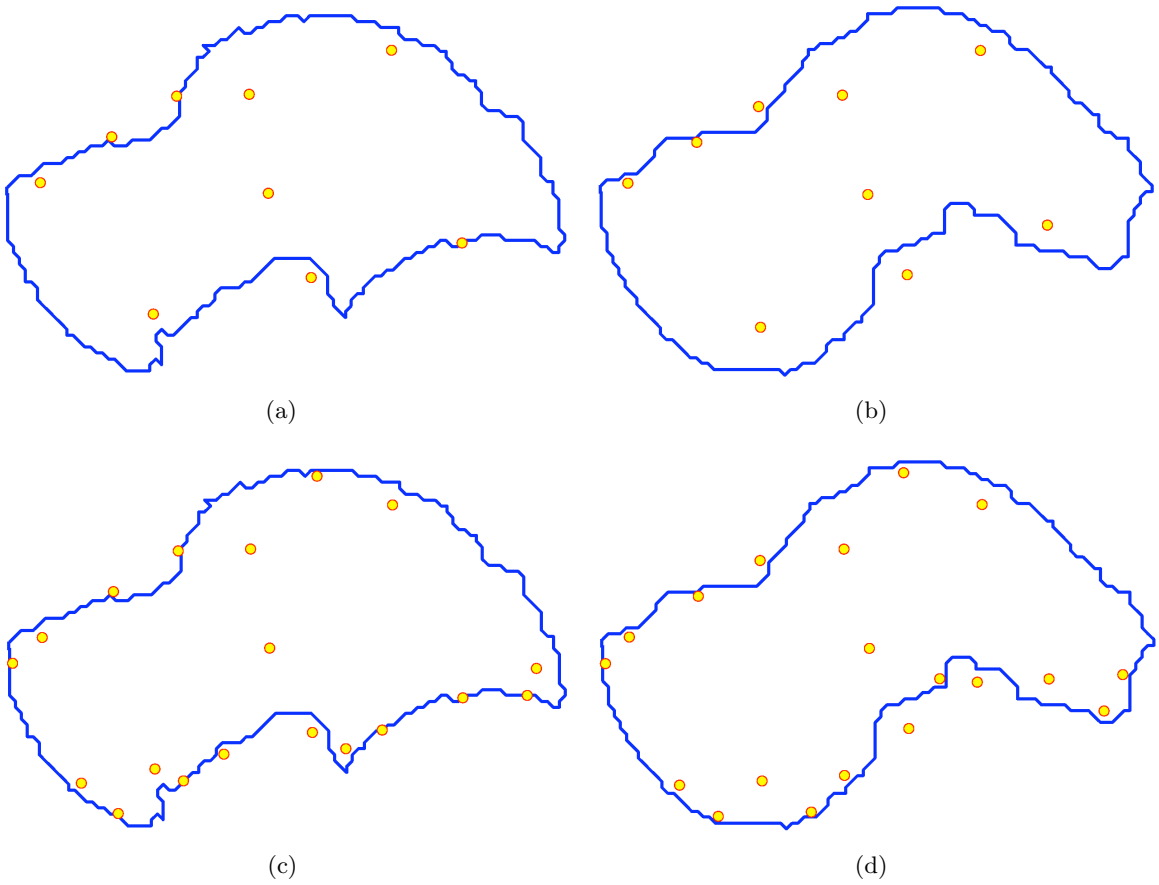


Figure 8.19: Virtual RBS landmark selection on two training shapes of the talus: (a-b) 10 landmarks, and (c-d) 20 landmarks. The centroids are selected as landmarks.

as landmarks the centroids obtained for each segment at each subdivision step. The higher the resolution, the closer the centroids come to the boundary, and the higher the level of detail obtained, as can be seen in Figs. 8.19 and 8.20 for the talus bone and the liver training set, respectively. The initial centroid is the centroid of the whole shape and is located at the

geometric centre of the shape for both examples presented.

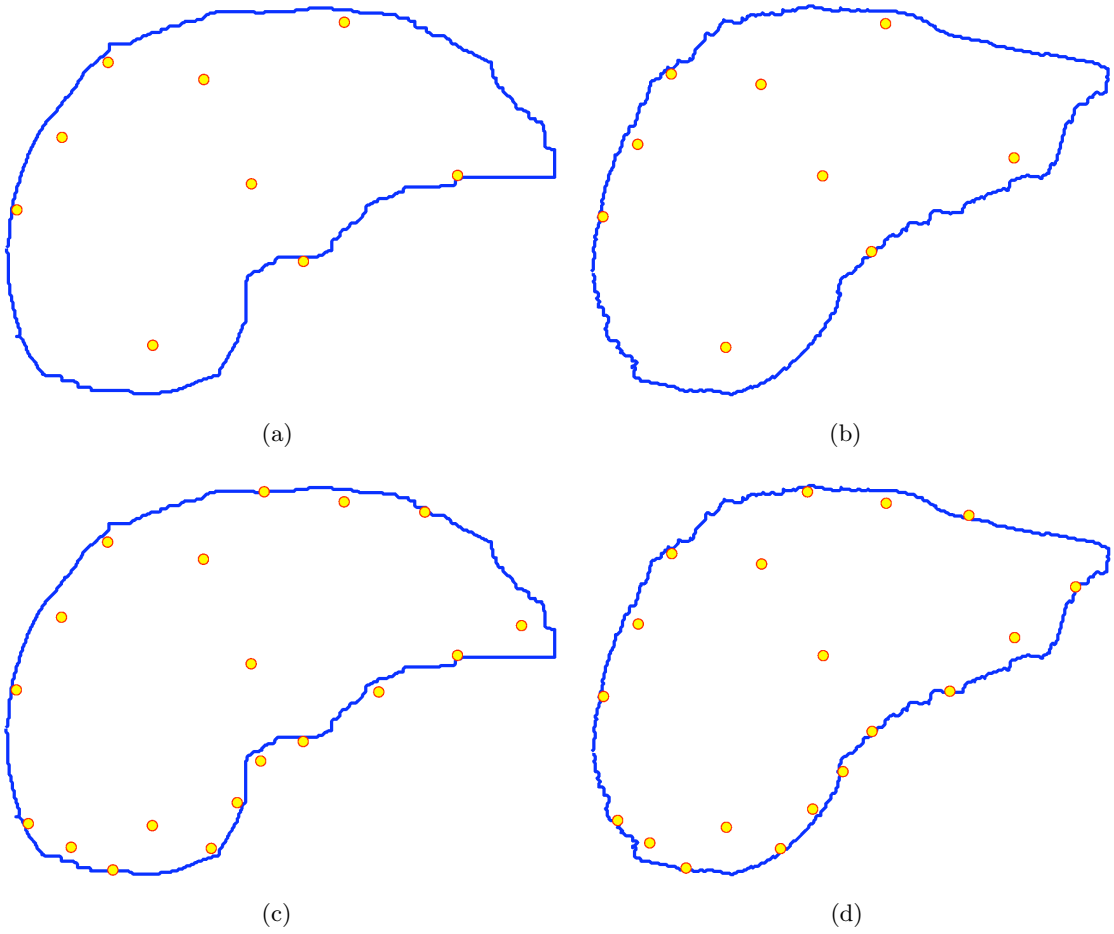


Figure 8.20: Virtual RBS landmark selection on two training shapes of the liver: (a-b) 10 landmarks, and (c-d) 20 landmarks. The centroids are selected as landmarks.

As for the other variants of the RBS family, the approach is multi-resolutional and as we increase the level of detail, it retains previous landmark positions.

Another strategy used as virtual landmarks consists of placing landmarks in the direction of the principal components, but at a certain distance from the centroid, proportional to the associated eigenvalue λ_1 or λ_2 (depending on the principal axis considered) by a factor a. The direction in which we place the landmarks is defined per convention, and it is assumed

to be always the direction pointing from the inside to the outside of each shape, this way we retain the landmark correspondence at each iteration. In Figs. 8.21 and 8.22, we show the resulting landmark selection for $a = 1$ and $a = 1.5$, respectively, for two distinct talus shapes and different number of landmarks.

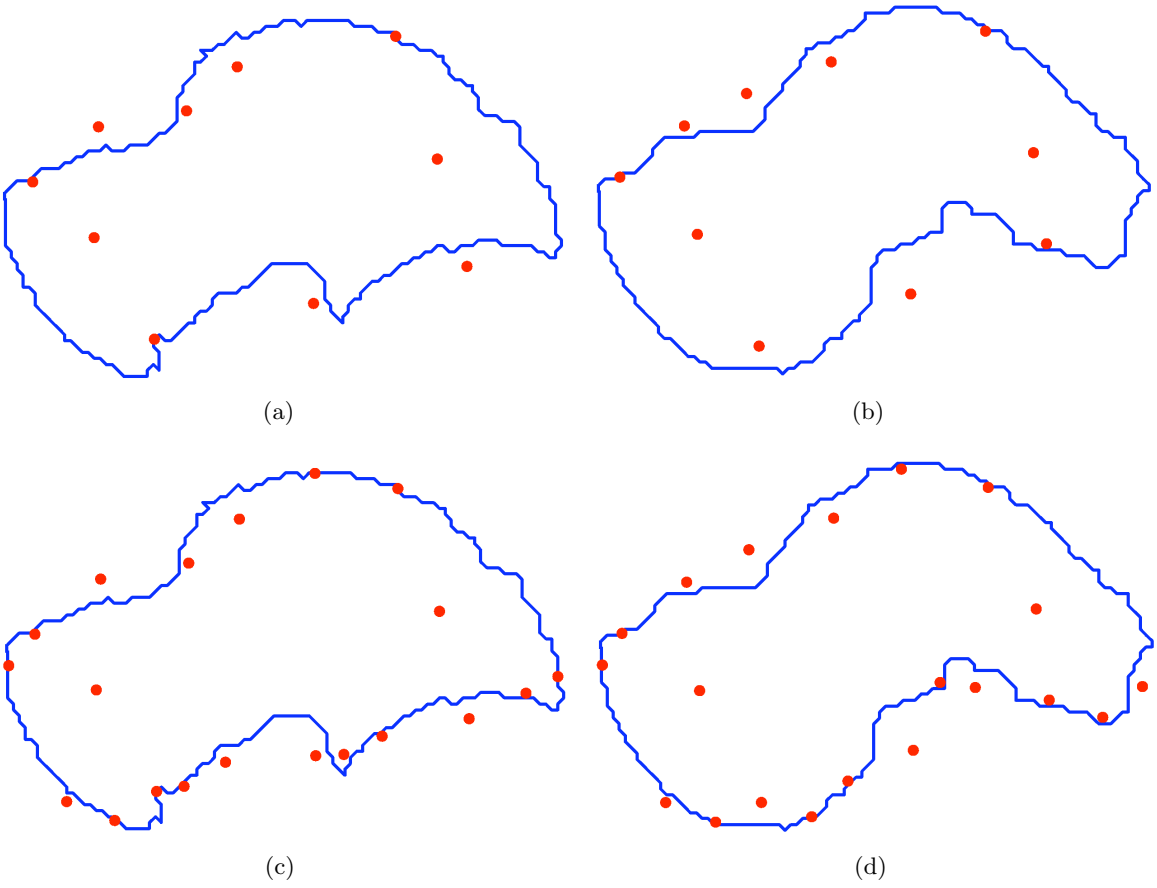


Figure 8.21: Virtual RBS landmark selection on two training shapes of the talus, with $a = 1$: (a-b) 10 landmarks, and (c-d) 20 landmarks.

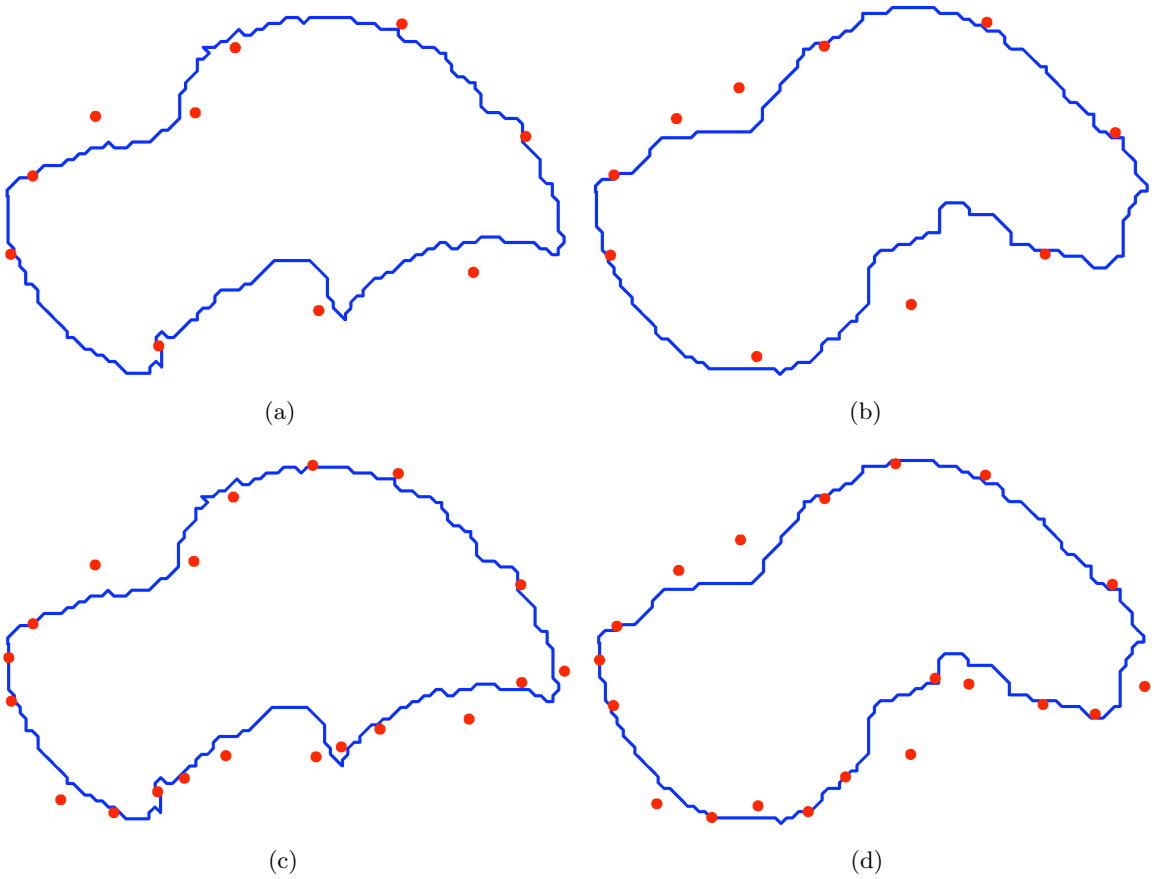


Figure 8.22: Virtual RBS landmark selection on two training shapes of the talus, with $a = 1.5$: (a-b) 10 landmarks, and (c-d) 20 landmarks.

The results obtained for the liver data set are shown in Figs. 8.23 and 8.24, for $a = 1$ and $a = 1.5$, respectively, for two distinct liver shapes and different number of landmarks.

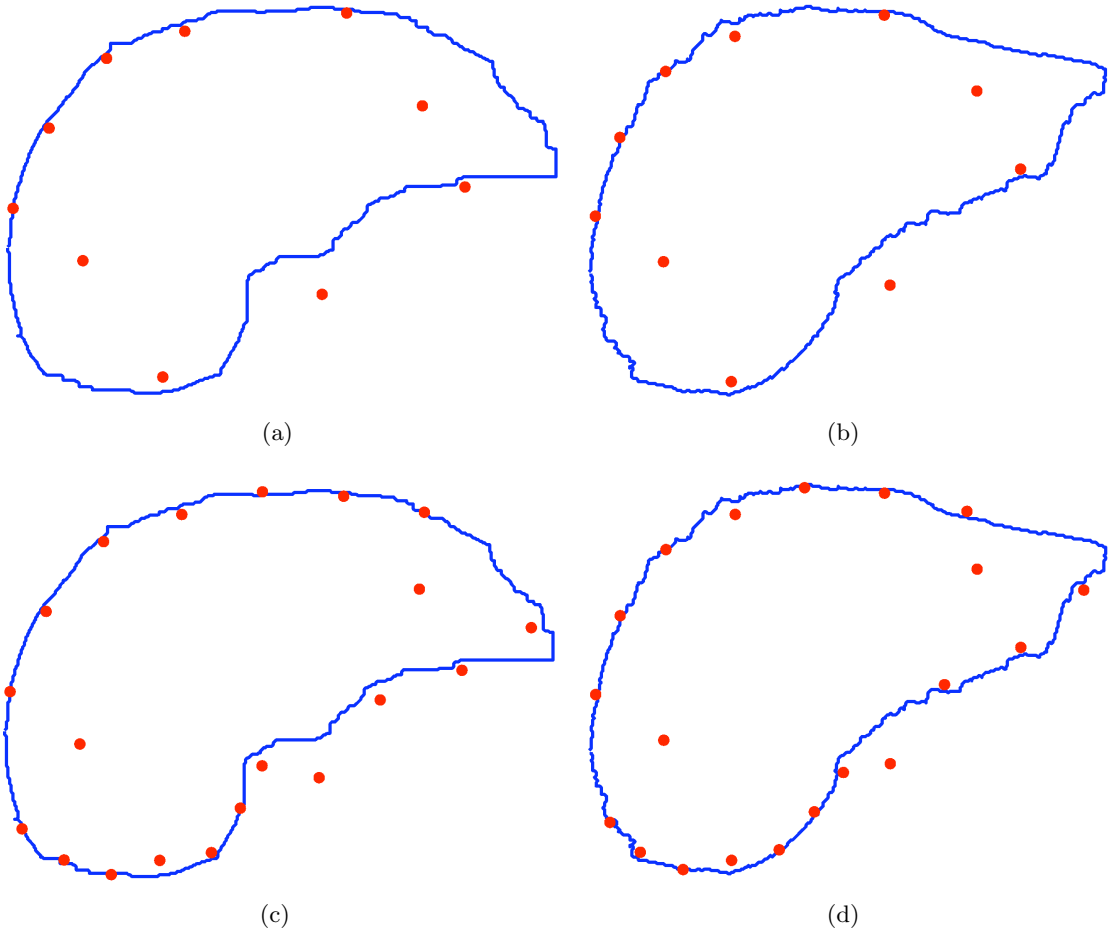


Figure 8.23: Virtual RBS landmark selection on two training shapes of the liver, with $a = 1$: (a-b) 10 landmarks, and (c-d) 20 landmarks.

We note that the conclusions reached for other variants of the RBS method are true also for virtual landmarks. Furthermore, the higher the value of factor a, the further away from the centre of the shape the landmarks will be. In particular, for $a = 1.5$ the virtual landmarks seem to fall around the boundary shape. We can see that the correspondence of landmarks seems to be consistent and robust for these variants as well as for PCA. Virtual landmarks are directly related to the shapes and can characterize them in a different way from what

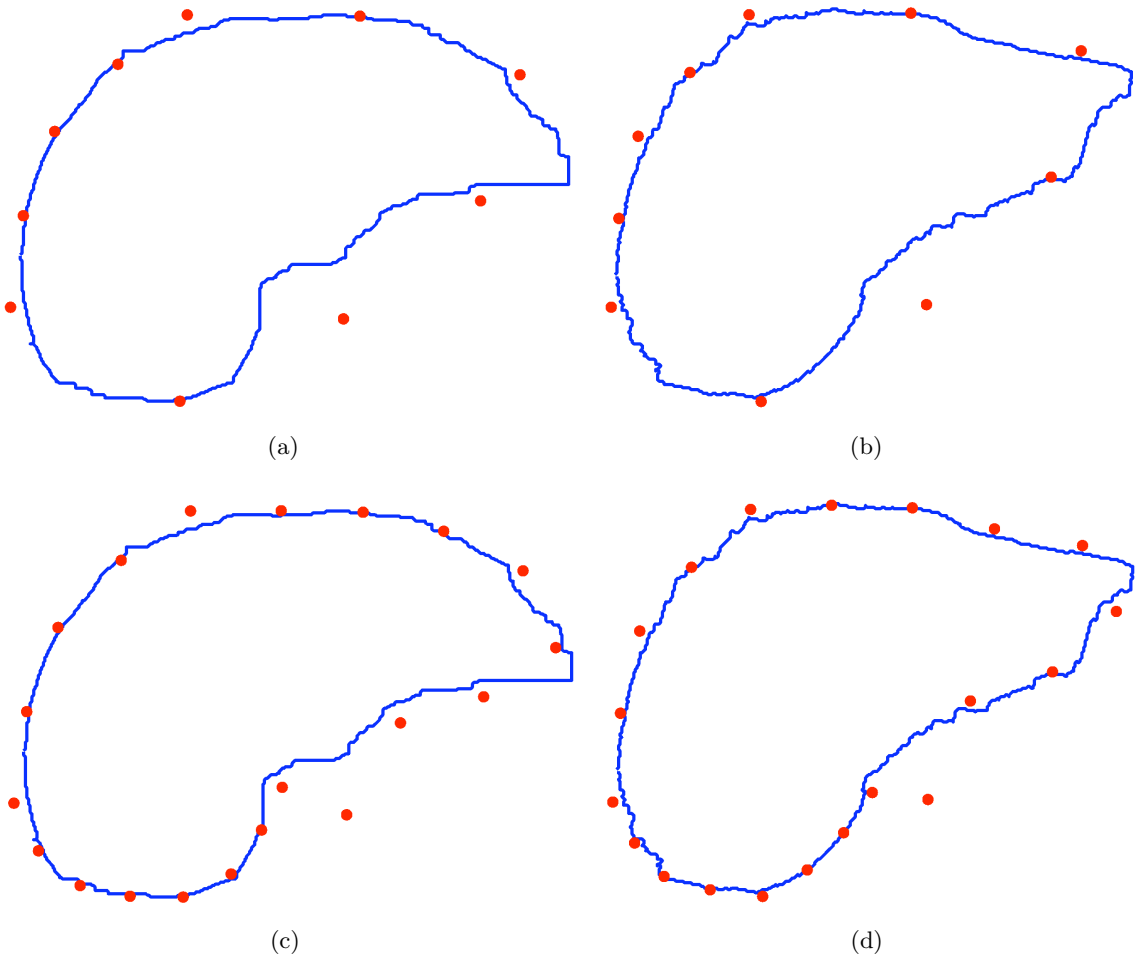


Figure 8.24: Virtual RBS landmark selection on two training shapes of the liver, with $a = 1.5$: (a-b) 10 landmarks, and (c-d) 20 landmarks.

has been considered in the literature up to now.

8.5 Quantitative Comparison

In this section, we compare all proposed landmark tagging methods with manual and equally spaced annotation methods in terms of compactness by using the compactness factors described in Section 8.2. In the following, we will calculate the compactness for each method separately, and compare it to manual and equally spaced strategies. In the end, we will compare all methods together and show which strategy is better for each of the two training

sets of shapes studied.

8.5.1 c-scale Based Landmark Tagging

In this section, we will use equations (8.2) and (8.3) to compare among the 12 methods, summarized in Section 5.7. Only sample comparative results will be presented and not an exhaustive listing of all evaluation results.

Hierarchical versus non-hierarchical selection

Fig. 8.25 shows a comparison of hierarchical and non-hierarchical landmark selection approaches, obtained on the talus training set, for $n = 1, ..., 18$, and for peaks and valleys separately. This plot shows that hierarchical selection creates more compact models than the

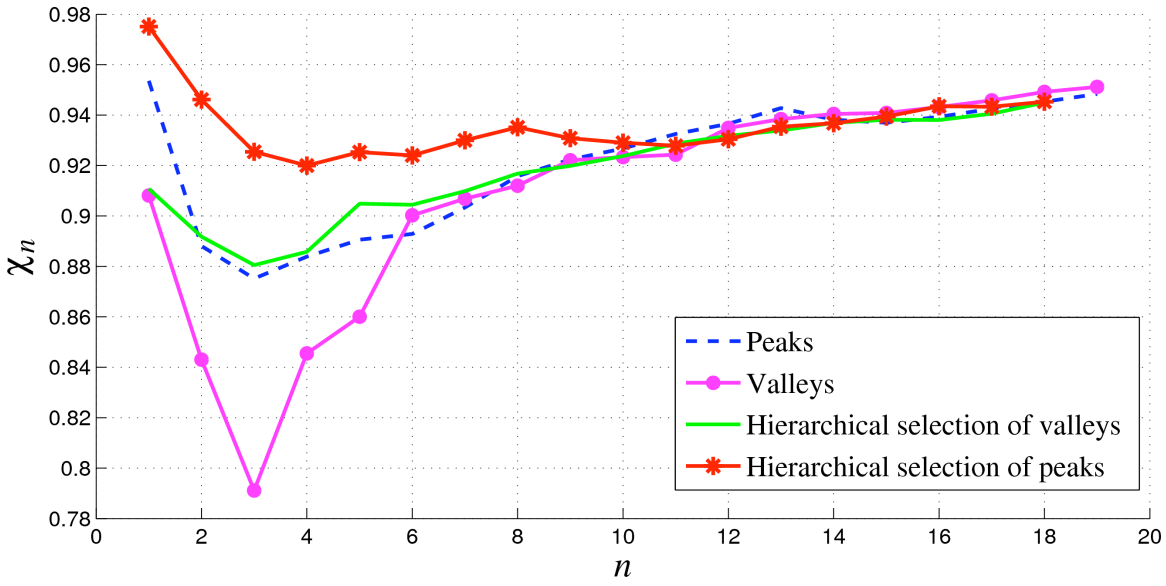


Figure 8.25: χ_n for $n = 1, ..., 18$. Comparison of hierarchical and non-hierarchical landmark tagging methods for the talus data set.

non-hierarchical approach, especially for small n. The hierarchical selection has also other advantages - it is easier to select landmarks in a hierarchical way, since the parameters are fixed in the beginning and do not need to be changed for each n. The global results for these two approaches compared to the manual and equally spaced methods are presented in terms

of χ in Table 8.1 for peaks and valleys for the talus bone. Overall, the hierarchical approach

Table 8.1: Comparison of hierarchical and non-hierarchical landmark tagging methods in terms of χ value for peaks and valleys separately in the talus data set.

	Non-hierarchical	Hierarchical	Equally spaced	Manual
Peaks	0.9204	0.9357	0.9095	0.9065
Valleys	0.9072	0.9190	0.9095	0.9065

performs better for both valleys and peaks than non-hierarchical, manual, and equally spaced methods. Note that for lower values of L, this difference in performance is further enhanced in favor of the hierarchical methods.

Closest point versus parametric propagation

We compare in terms of compactness the two different types of propagation introduced in Section 5.5. From Fig. 8.26, we may observe that the parametric approach performs better

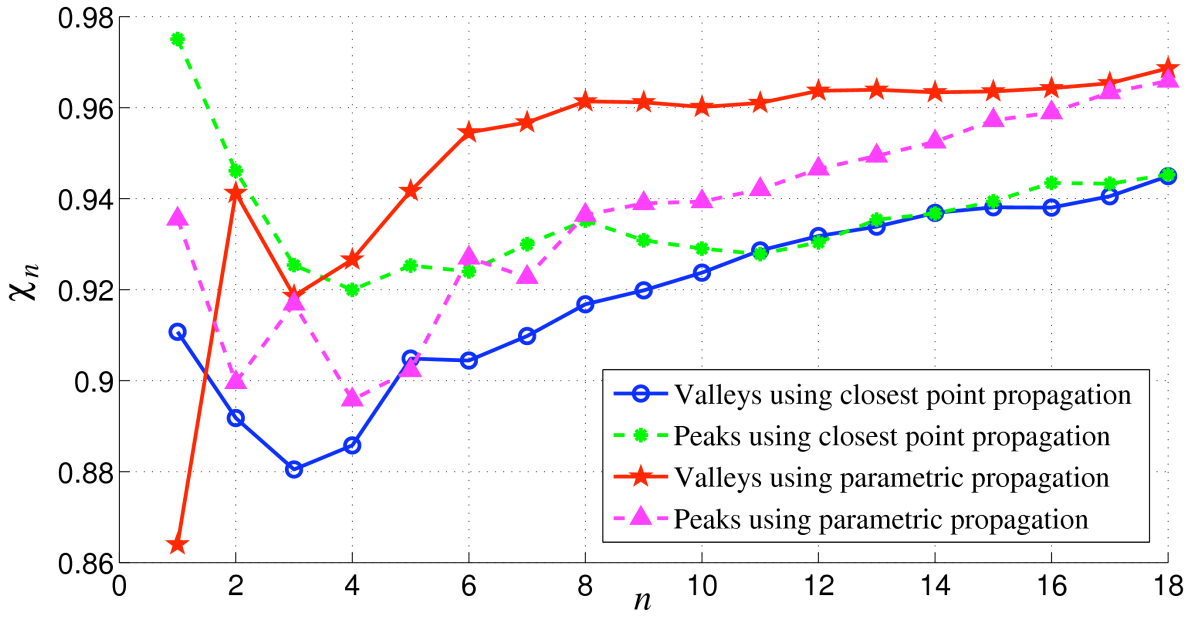


Figure 8.26: χ_n for $n = 1, ..., 18$. Comparison of closest point and parametric propagations for the talus data set.

for valleys in the talus data set. For peaks, it performs better for $n > 8$. For a small number

of peaks selected, the closest point method gives better results. The results in terms of χ are listed in Table 8.2. For peaks and valleys, parametric propagation performs better than

Table 8.2: Comparison of closest point and parametric propagation in terms of χ value for peaks and valleys separately for the talus data set.

	Closest point	**Parametric**	**Equally spaced**	**Manual**
Peaks	0.9357	0.9362	0.9095	0.9065
Valleys	0.9190	0.95	0.9095	0.9065

closest point propagation in the talus data set. Both methods for peaks and valleys give better results than manual and equally spaced methods.

The results for these two propagation approaches when we select both peaks and valleys for $n = 1, ..., 27$, are displayed via χ_n in Fig. 8.27. If we consider both peaks and valleys

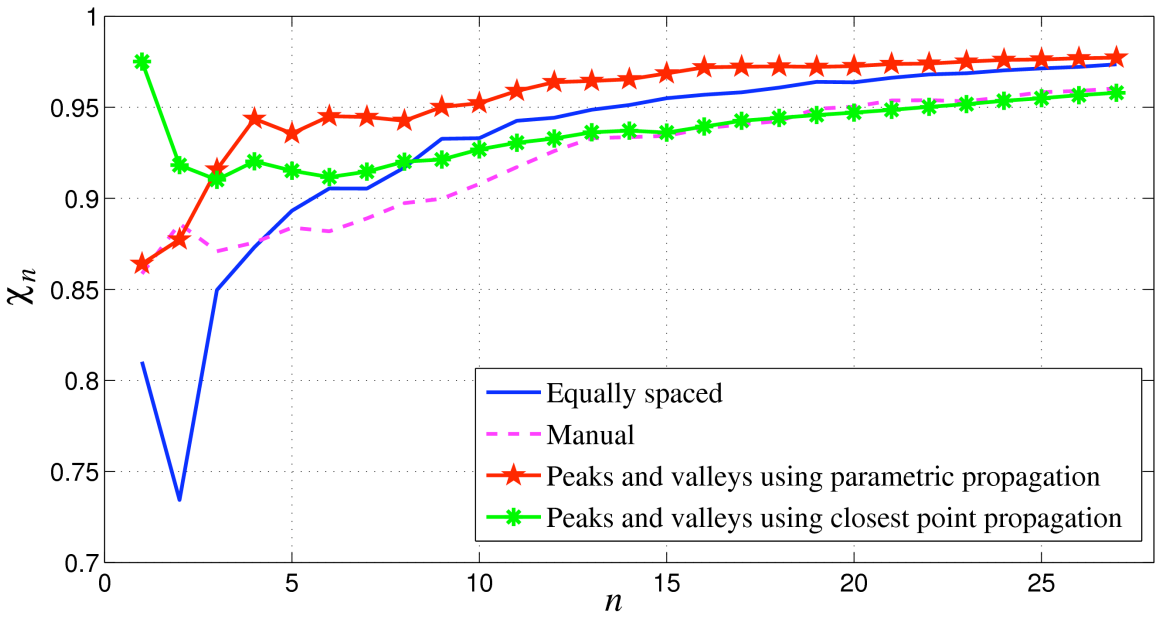


Figure 8.27: χ_n for $n = 1, ..., 27$. Comparison of closest point and parametric propagations for the talus data set.

(half peaks and half valleys), then, the parametric propagation of landmarks is more suitable to obtain a compact model for the talus data set. This is also presented numerically

in Table 8.3. Again, both methods of propagating landmarks perform globally better than

Table 8.3: Comparison of closest point and parametric propagation in terms of χ value for peaks and valleys for the talus data set.

	Closest point	**Parametric**	**Equally spaced**	**Manual**
χ	0.9370	0.9549	0.9293	0.9226

manual and equally spaced approaches, but the parametric approach gives the best results. We may also observe in Fig. 8.27 that only the parametric propagation curve is above both manual and equally spaced plots. This means that this type of propagation gives better results independently of the number of landmarks used, whereas closest point propagation is better than the equally spaced and manual methods only for small n.

The results for the liver data set, with these two propagation approaches, when we select both peaks and valleys for $n = 1, ..., 30$, are displayed via χ_n in Fig. 8.28. If we consider

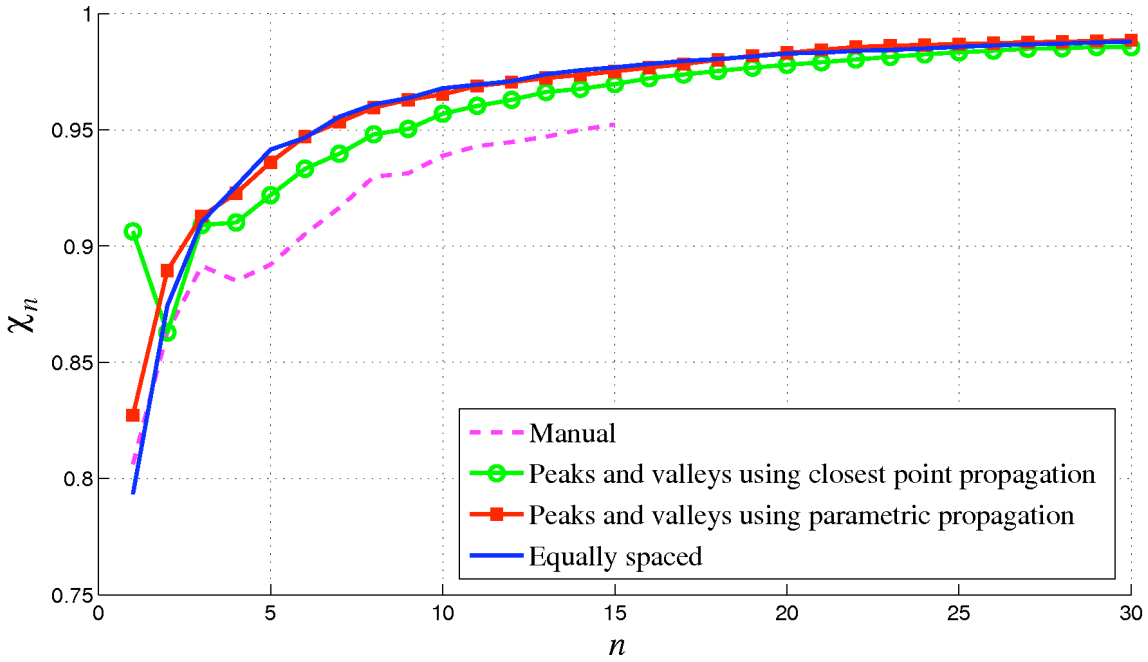


Figure 8.28: χ_n for $n = 1, ..., 30$. Comparison of closest point and parametric propagations for the liver data set.

both peaks and valleys (half peaks and half valleys), then, the parametric propagation of landmarks is more suitable to obtain a compact model for the liver. This is also presented numerically in Table 8.4. In this case, parametric propagation performs globally better than

Table 8.4: Comparison of closest point and parametric propagation in terms of χ value for peaks and valleys for the liver data set.

	Closest point	**Parametric**	**Equally spaced**
χ	0.9591	0.9635	0.9623

the equally spaced approach. In Fig. 8.28, the parametric propagation performs similarly to the equally spaced approach, but better than the closest point propagation independently of the number of landmarks used. Both propagation methods perform better than the manual approach. Overall, for the liver and according to Table 8.4, parametric propagation performs better than equally spaced and closest point propagation.

We may conclude that, in the two training sets considered, parametric propagation performs better than closest point propagation, and also than manual and equally spaced approaches overall.

Hierarchical selection and parametric propagation

In this part, we show the results for $n = 1, ..., 18$ considering peaks only, valleys only, and valleys and peaks together in the talus data set. The selection of landmarks is done hierarchically and the propagation is achieved parametrically. The method with both peaks and valleys was obtained by using, for each number of landmarks, half number of peaks and half of valleys. The results are displayed in Fig. 8.29. For the talus data set, the three methods (valleys, peaks, and both) perform better than manual and equally spaced methods, the best results being for valleys and valleys and peaks together. The global result for this experiment is listed in Table 8.5. Globally, the method that performs better for the talus data set is the one that considers only valleys, followed by the one that considers valleys and peaks together. All three methods perform better than equally spaced and manual landmark tagging

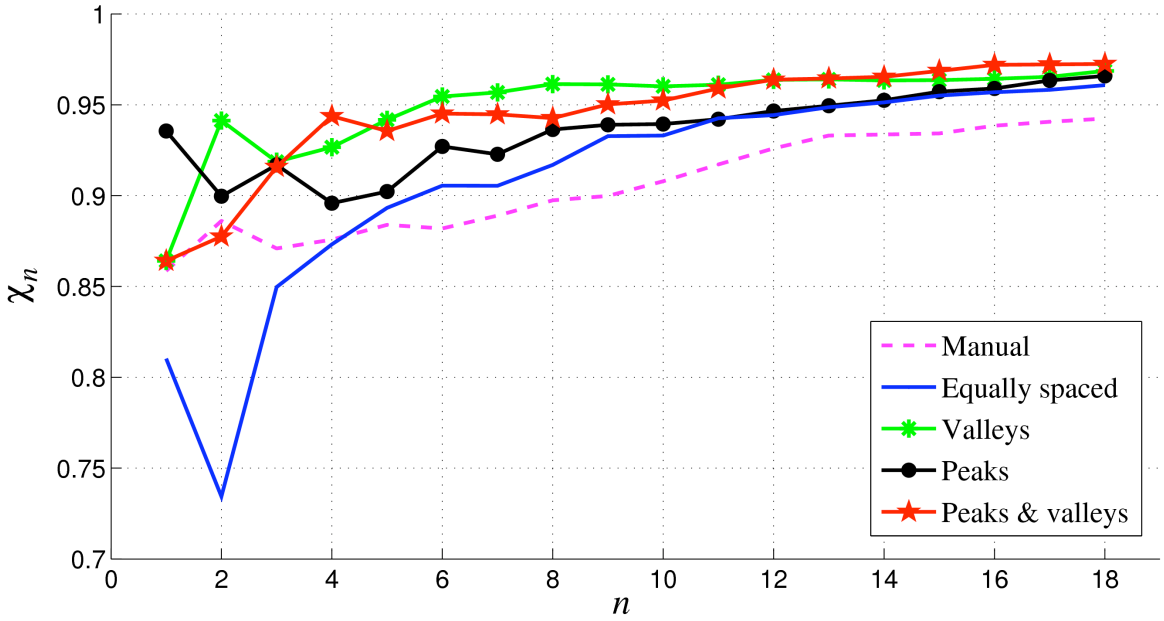


Figure 8.29: χ_n for $n = 1, ..., 18$. Hierarchical selection of landmarks and parametric propagation for the talus data set.

Table 8.5: Comparison of methods in terms of χ value for peaks and valleys for the talus data set.

	Peaks	**Valleys**	**Peaks & valleys**	**Equally spaced**	**Manual**
χ	0.9362	0.95	0.9450	0.9095	0.9065

approaches.

In the liver data set, the results for $n = 1, ..., 15$ considering peaks only, valleys only, and valleys and peaks together, with the selection of landmarks done hierarchically and the propagation achieved parametrically, are displayed in Fig.8.30 and Table 8.6. The method with both peaks and valleys was obtained by using, for each number of landmarks, half number of peaks and half of valleys. The three methods (valleys, peaks, and both) perform better in terms of χ_n than manual and equally spaced methods, the best results being for peaks only. Valleys and peaks together perform similarly to the equally spaced method, but better than

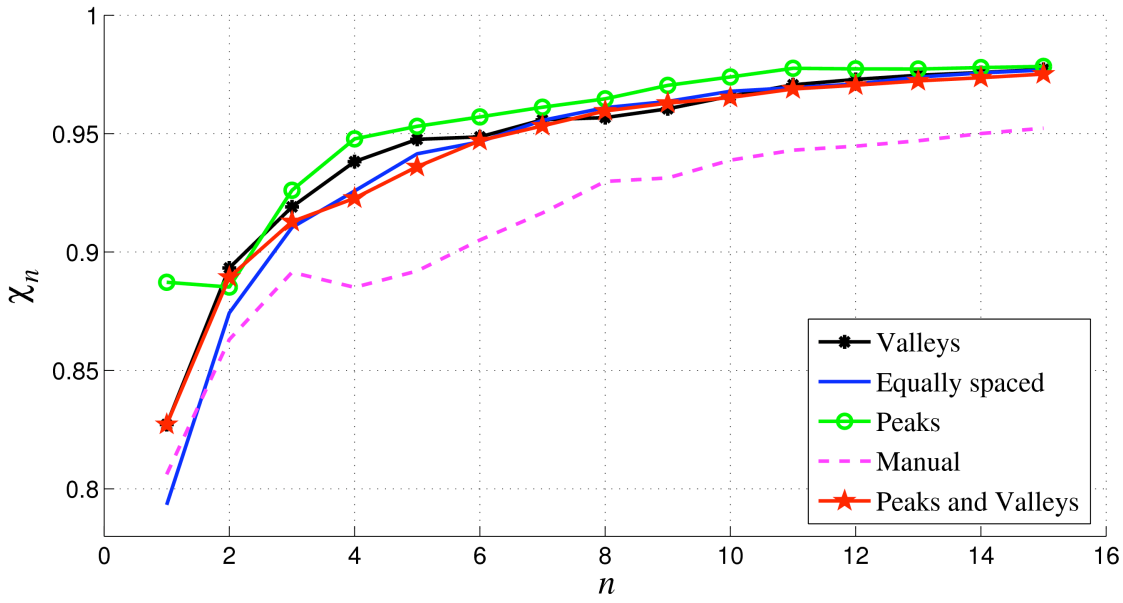


Figure 8.30: χ_n for $n = 1, ..., 15$. Hierarchical selection of landmarks and parametric propagation for the liver data set.

the manual approach.

Table 8.6: Comparison of methods in terms of χ value for peaks and valleys for the liver data set.

	Peaks	**Valleys**	**Peaks and valleys**	**Equally spaced**	**Manual**
χ	0.9544	0.9456	0.9424	0.9405	0.9131

Overall in the liver data set and in terms of χ, the proposed methods perform better than equally spaced and manual approaches, the best result obtained by using peaks as landmarks. If we observe the mean shape of the liver, fewer high curvature points or corners can be located, but more straight segments are present, than in the talus. This may be one of the reasons why peaks represent a better value in the results. This also demonstrates that the most appropriate use of peaks, valleys, or both as landmarks, may depend on the application.

8.5.2 Variance Equalization

We compare the variance equalization strategy with the manual and equally spaced annotations in terms of χ_n value, for $n = 1, ..., 18$, as shown in Fig. 8.31 for the talus bone.

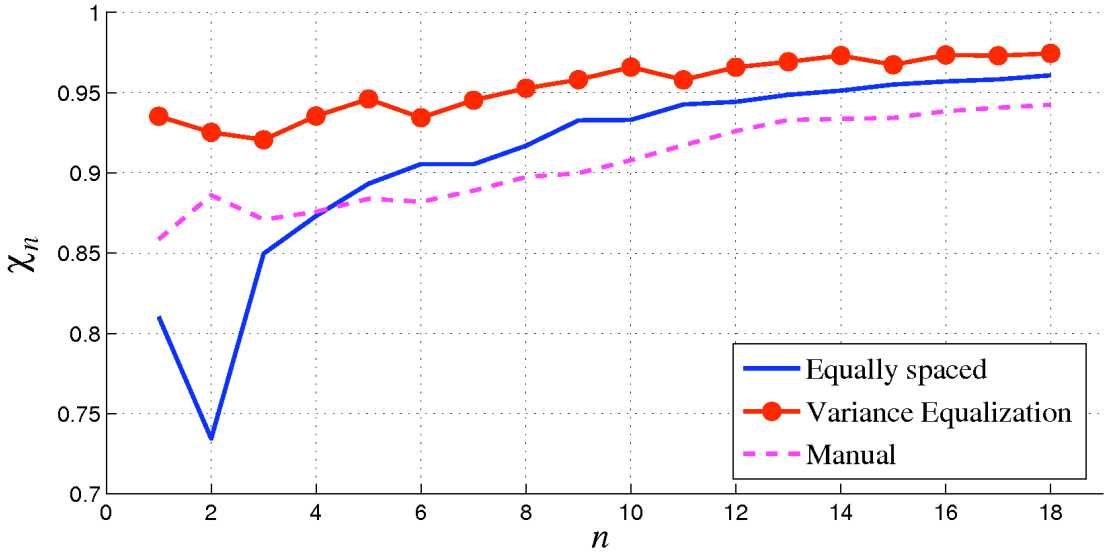


Figure 8.31: χ_n for $n = 1, ..., 18$. Selection of landmarks via variance equalization for the talus data set.

The variance equalization curve is closer to $\chi_n = 1$, which means that it is more compact than the other two methods, therefore, it seems to perform significantly better than manual and equally spaced approaches for all number of landmarks considered in the talus training set. The results in terms of χ value for the talus data set, for $n = 1, ..., 18$, are presented in Table 8.7.

Table 8.7: Comparison of methods in terms of χ value for the talus data set.

	Variance Equalization	**Equally spaced**	**Manual**
χ	0.9540	0.9095	0.9065

For the talus, we can conclude that the variance equalization method performs better than manual and equally spaced approaches in terms of χ_n and χ values.

For the liver data set, the results of compactness for different number of landmarks, $n = 1, ..., 15$, are displayed via χ_n in Fig. 8.32 and in terms of χ value in Table 8.8.

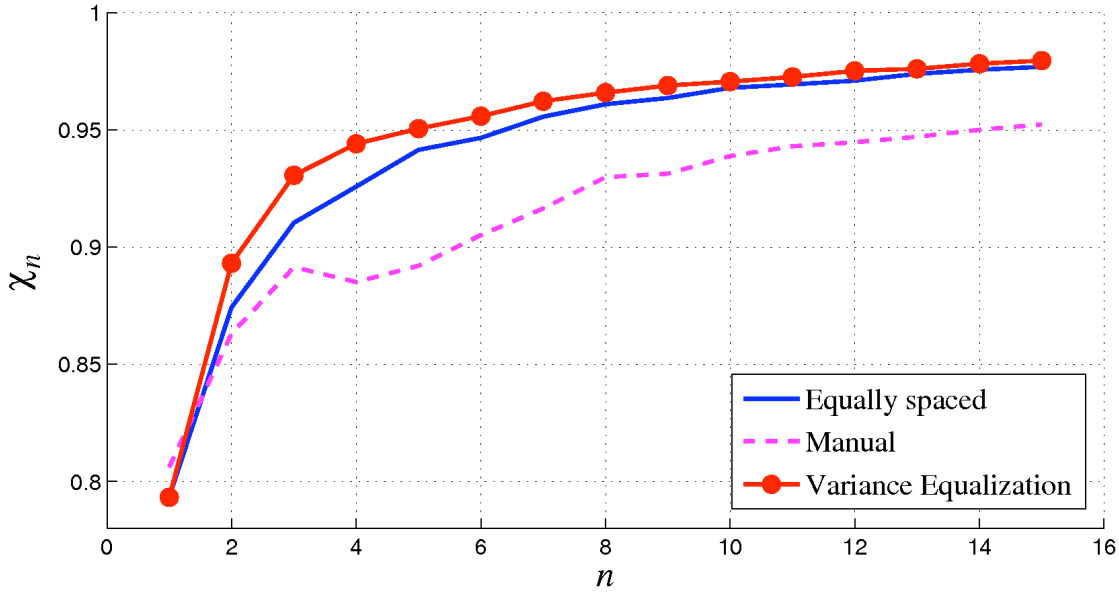


Figure 8.32: χ_n for $n = 1, ..., 15$. Selection of landmarks via variance equalization for the liver data set.

For the liver, the variance equalization approach performs better at all number of landmarks in terms of χ_n and χ values.

Table 8.8: Comparison of methods in terms of χ value for the liver data set.

	Variance Equalization	**Equally spaced**	**Manual**
χ	0.9478	0.9405	0.9131

All these results show that the variance equalization method performs better in terms of compactness than the equally spaced and the manual methods for both shapes studied. Therefore, distributing the landmarks according to the total variance present on a training set creates a more compact model to build PDM for model-based segmentation methods such as those using ASMs. It does not depend on a starting point as does the equally spaced method, and it avoids the difficulties and drawbacks of manual annotation.

8.5.3 Recursive Boundary Subdivision

We compared all variants presented for the RBS method with the manual and equally spaced annotations in terms of compactness for the talus bone and for the liver. In Fig. 8.33, we display the results for the talus data set, for $n = 1, ..., 18$, via χ_n.

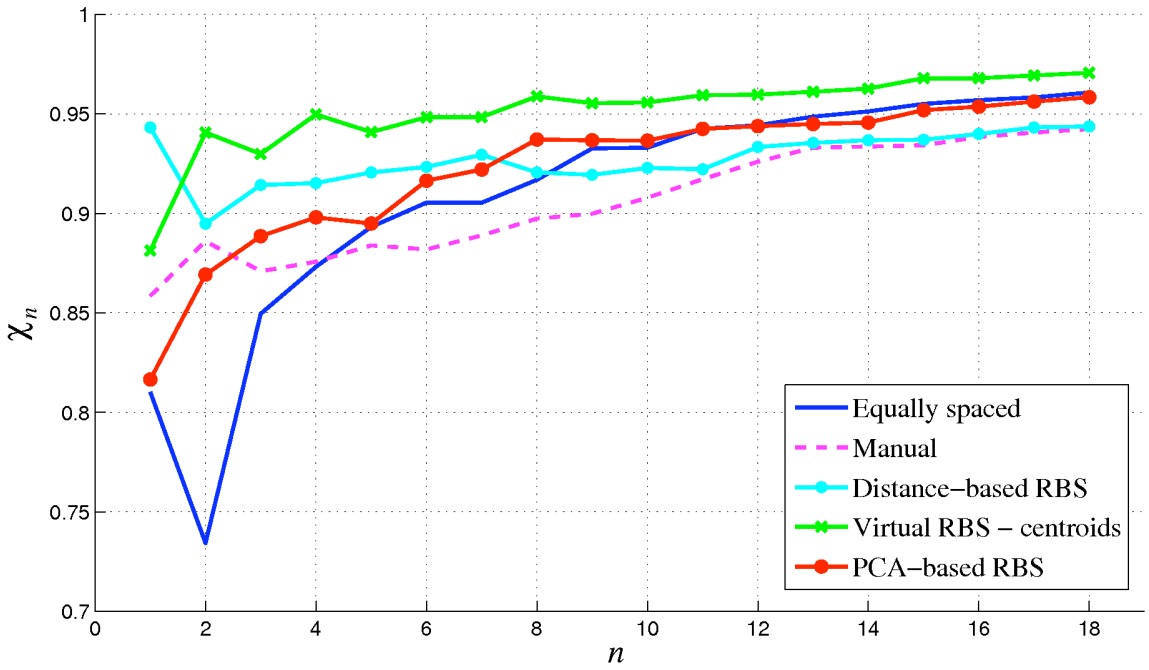


Figure 8.33: χ_n for $n = 1, ..., 18$. Selection of landmarks via RBS for the talus data set.

According to Fig. 8.33, the method that is closer to $\chi_n = 1$ is the virtual RBS based on the centroids. This variant seems to perform better than any other method for any number of landmarks greater than 1. The PCA-based RBS is more compact than the manual and equally spaced annotations, although it is close to the equally spaced annotation curve for high number of landmarks ($n > 9$). The distance-based RBS performs better than equally spaced and manual methods only for a number of landmarks smaller than 8. For higher values it has a lower χ_n value than the equally spaced approach, but still higher than the manual method, although for a number of landmarks greater than 13, the curve is very similar to the manual annotation plot.

The results in terms of χ value are summarized in Table 8.9. For the talus training set,

Table 8.9: Comparison of RBS methods in terms of χ value for the talus data set.

	Dist. RBS	**PCA-RBS**	**Virtual RBS**	**Equally spaced**	**Manual**
χ	0.9275	0.9230	0.9516	0.9095	0.9065

considering $L = 18$, the best strategy seems to be the virtual RBS based on centroids, followed by the distance-based RBS and the PCA-RBS. All the variants perform better than manual and equally spaced methods for this particular application.

The virtual RBS based on eigenvalues could not be compared to the other methods since the compactness varies depending on the parameter a selected. To investigate this further, in Fig. 8.34 we study how this parameter affects the compactness factor χ. As can be observed,

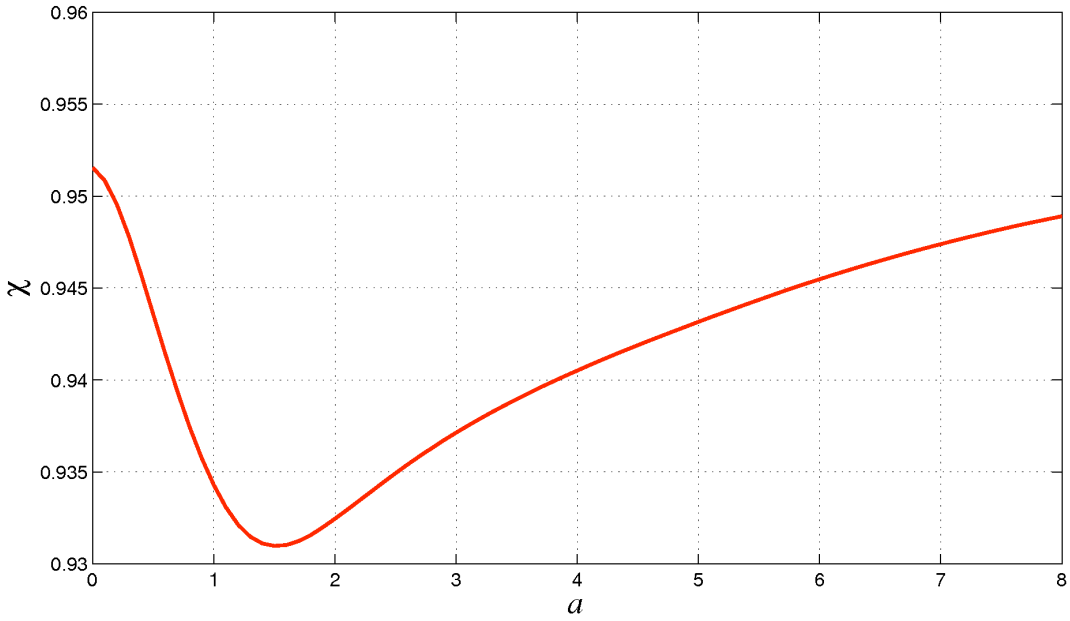


Figure 8.34: χ as a function of a. Selection of landmarks via virtual RBS based on the eigenvalues for the talus data set.

χ varies considerably depending on the factor a selected, reaching a minimum for $a = 1.5$ in this particular case. Surmising that this curve may depend on shape complexity, we applied

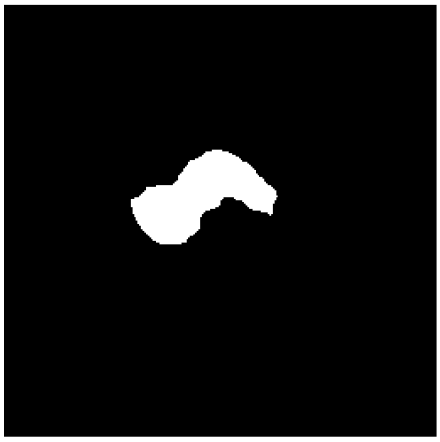
erosions and dilations by using a circular structuring element (se) to the original training set, to simplify or increase the shape complexity of the talus data set and studied how this affected χ. We applied to all 40 shapes of the talus data set two erosions with $se = 3$ and $se = 4$ and three dilations with $se = 5$, $se = 10$, and $se = 15$, to see how the shape complexity changed; one shape from this family is presented in Fig. 8.35. Eroding the talus shape seems to smooth the original shape a bit but in a smaller proportion than by dilation. Sometimes, erosion can increase shape complexity. For the talus, certain areas get smoothed by erosion, such as the cavity at the bottom part of the shape. The higher the dilation applied to the talus shape, the smoother and rounded shape we obtain. In Fig. 8.36, we present χ as a function of a for all newly generated shapes.

From Fig. 8.36, it appears that as the shape becomes smoother, higher compactness is obtained. For highly dilated shapes, the results seem to be more compact than for eroded and original shapes. In this case, when the dilation is small ($se = 5$), the results obtained are similar to the curves obtained for eroded shapes. For the eroded shapes, the changes with respect to the original shape were not so pronounced as for the dilated shapes, this is why the curves corresponding to the eroded shapes are lower than the ones associated with the dilation. It is also interesting to note that, for the original shape, the highest compactness was obtained at $a = 0$, which means that landmarks were located at the centroids of each segment. However, when we start smoothing the shapes, the curve changes and the pitting becomes less deep, whereas the maximum compactness obtained at higher values of a increases, and reaches a χ value higher than the one obtained at $a = 0$. Furthermore, the smoother the shape, the quicker the curve reaches the minimum, and the faster it increases subsequently.

A similar comparison for all the variants of the RBS family was carried out for the liver training set, for $n = 1, ..., 15$, as presented in Fig. 8.37. In this case, the best compactness result seems to occur for PCA-based RBS, followed by virtual RBS based on the selection of centroids. Distance-based RBS performs similarly to the equally spaced method. They all

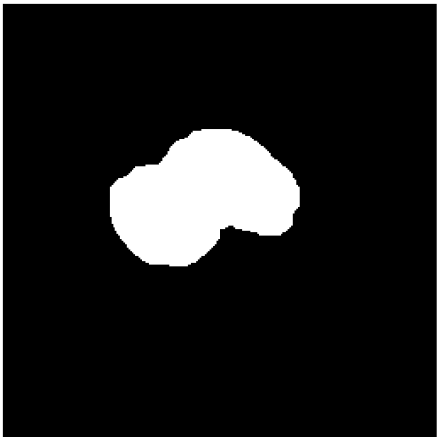
(a) Original.

(b) Erosion $se = 3$.

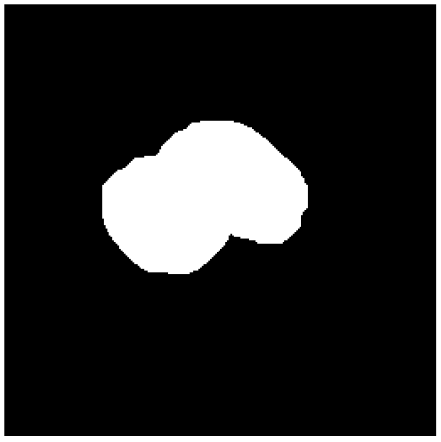


(c) Erosion $se = 4$.

(d) Dilation $se = 5$.



(e) Dilation $se = 10$.



(f) Dilation $se = 15$.

Figure 8.35: Examples of erosion and dilation, with different sizes of se, applied to the talus bone shape in (a).

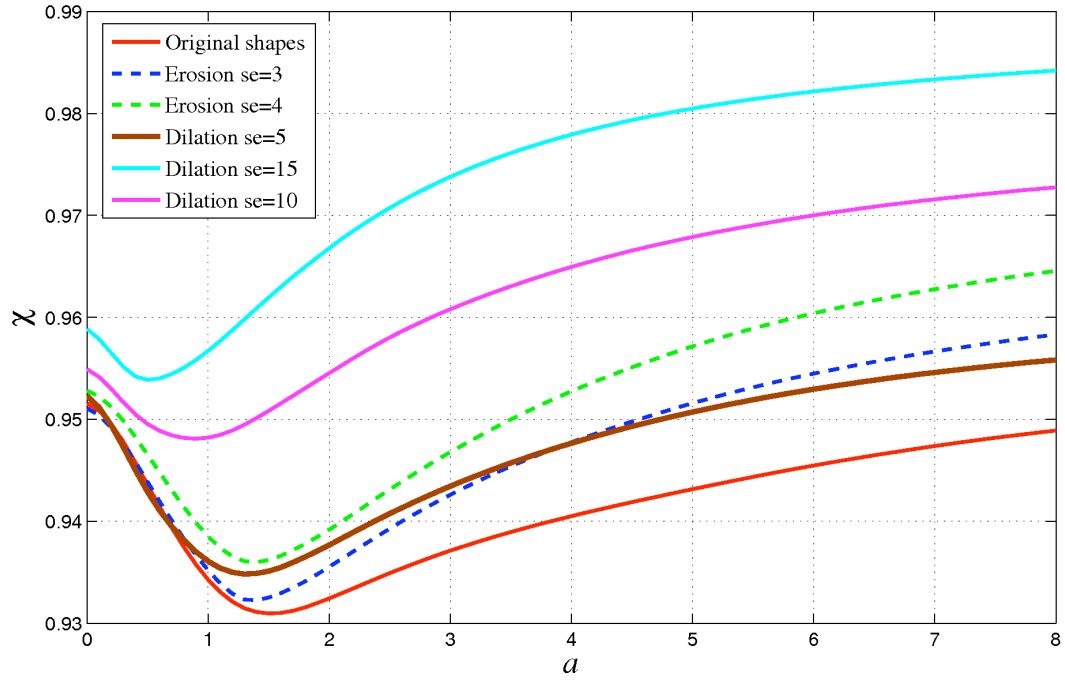


Figure 8.36: χ as a function of a. Selection of landmarks via virtual RBS based on the eigenvalues for the talus data set as shape complexity is changed.

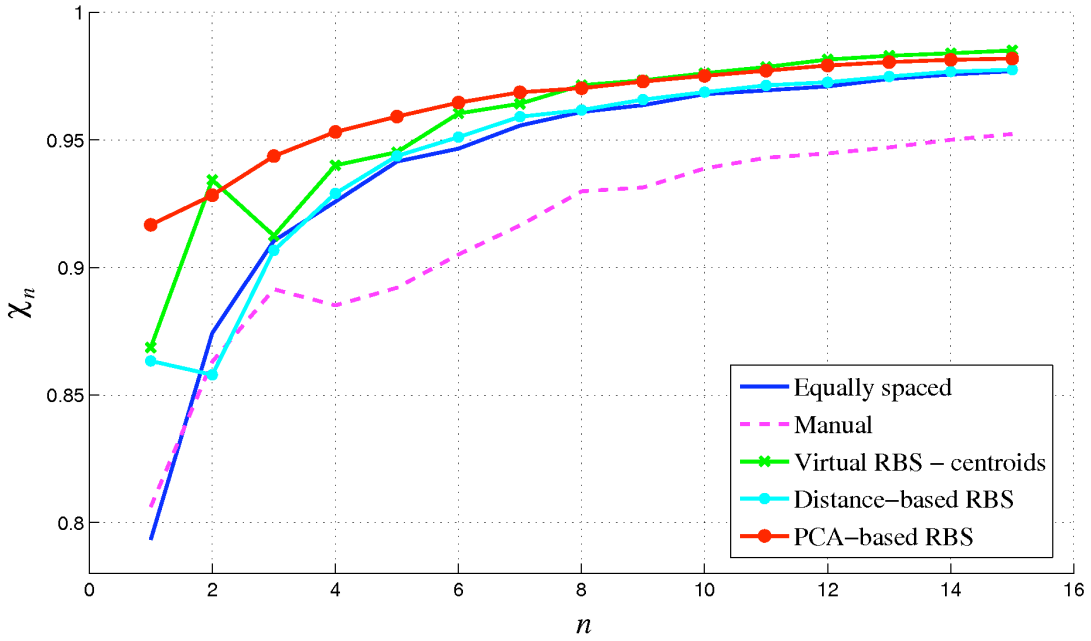


Figure 8.37: χ_n for $n = 1, ..., 15$. Selection of landmarks via RBS for the liver data set.

perform better than manual annotation in terms of χ_n. A similar behaviour can be observed for the χ factor in Table 8.10. Overall for the liver, PCA-based RBS creates more compact

Table 8.10: Comparison of RBS methods in terms of χ value for the liver data set.

	Dist. RBS	**PCA-RBS**	**Virtual RBS**	**Equally spaced**	**Manual**
χ	0.9453	0.9635	0.9572	0.9405	0.9131

models, followed by virtual RBS based on the centroids, and by distance-based RBS, which still performs better than the equally spaced and manual strategies.

To analyse the behaviour of virtual RBS based on eigenvalues, we calculated the overall compactness value χ for different values of a as shown in Fig. 8.38. In this case, the min-

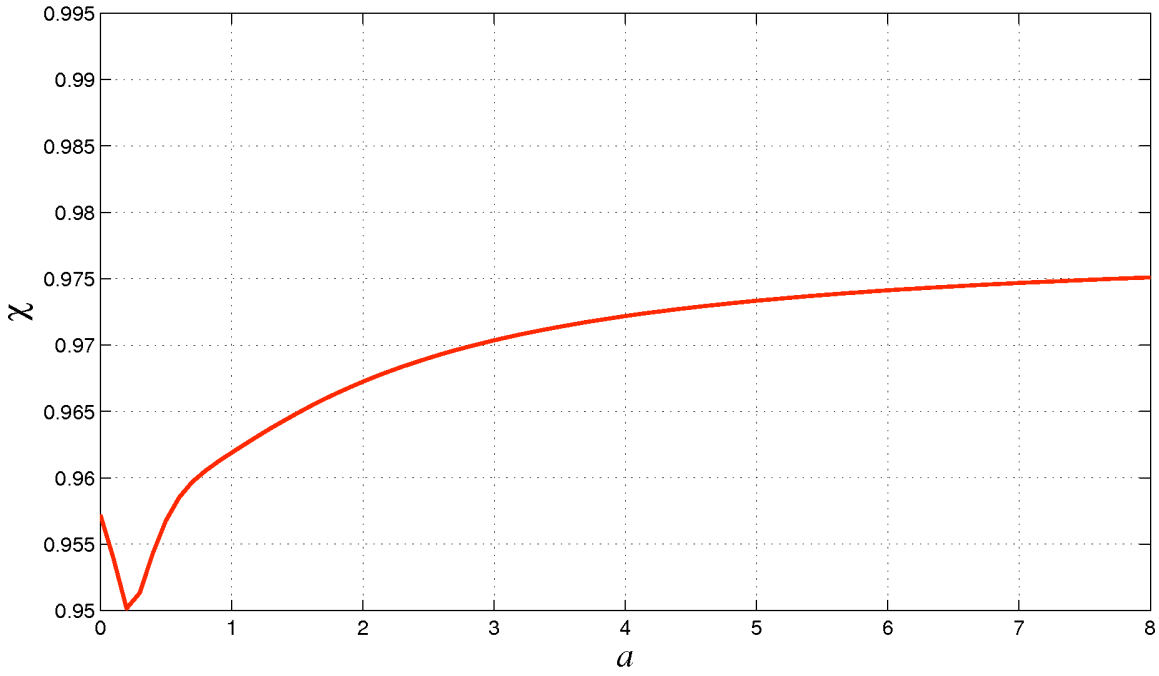
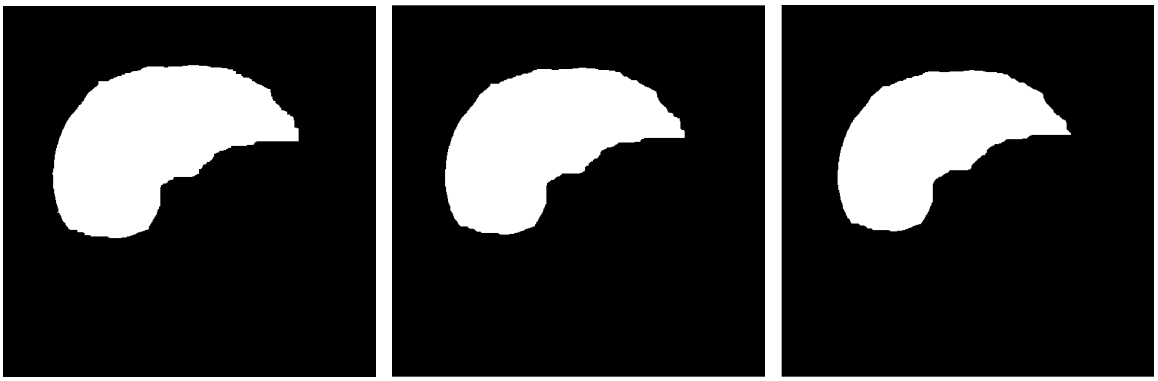


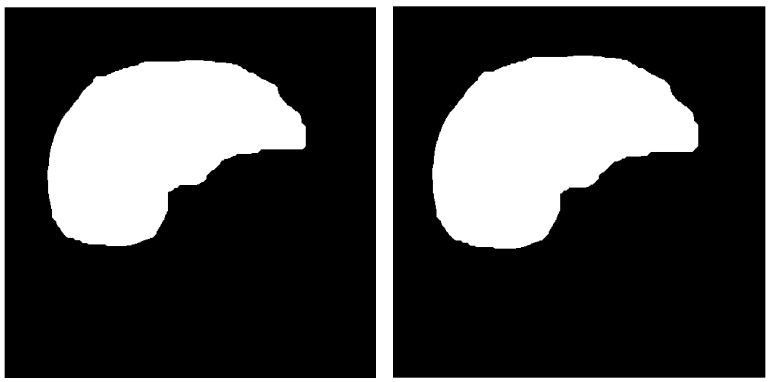
Figure 8.38: χ as a function of a. Selection of landmarks via virtual RBS based on eigenvalues for the liver data set.

imum of the curve is obtained at smaller a values than for the talus, and the compactness increases after that point reaching higher values than those obtained at $a = 0$ even for small

values of a. To study this with respect to the changes in shape complexity, we created a set of training sets of liver shapes with different complexity levels by eroding and dilating the original liver training set. One sample of the original training set along with its eroded and dilated versions are shown in Fig. 8.39.



(a) Original. (b) Erosion $se = 5$. (c) Erosion $se = 9$.



(d) Dilation $se = 10$. (e) Dilation $se = 15$.

Figure 8.39: Examples of erosion and dilation with different sizes of se applied to the liver shape in (a).

The results of χ as a function of a for these shapes are presented in Fig. 8.40. We notice that the compactness factor χ calculated over a range of a values, is different for different shape complexities. In this case, the compactness seems to be better for the eroded shapes and worse for the dilated shapes. The minima of the curves, for all shapes considered, seem to happen at a similar value of a.

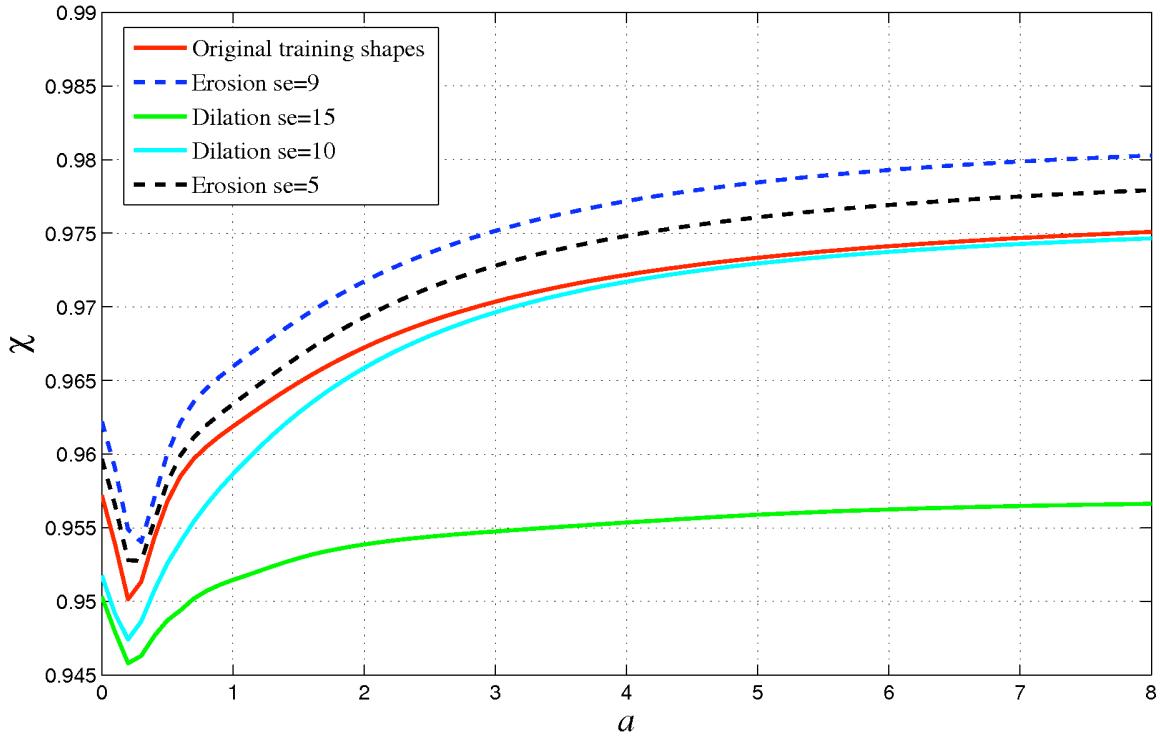
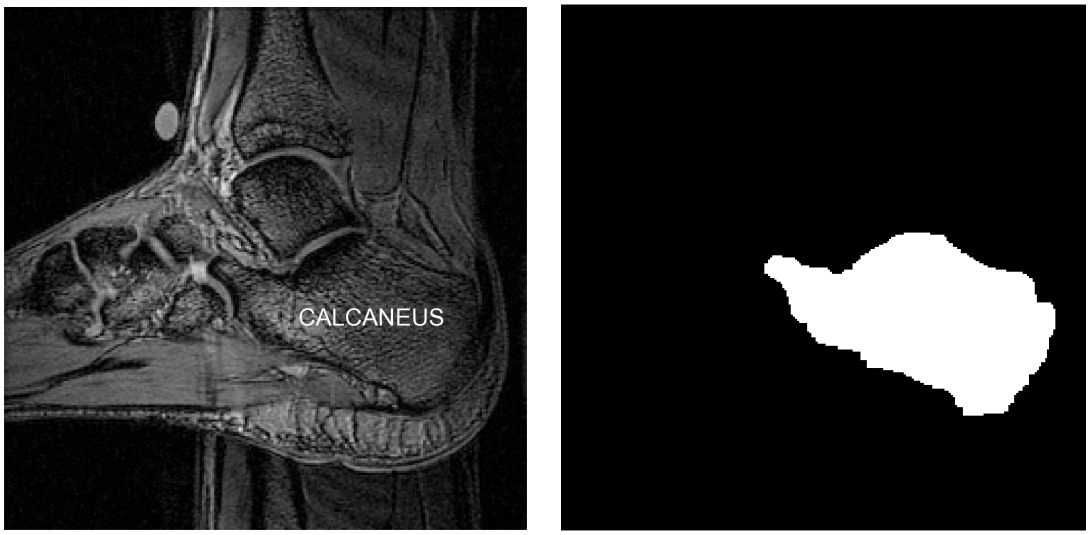


Figure 8.40: χ as a function of a. Selection of landmarks via virtual RBS based on eigenvalues for the liver data set as shape complexity is changed.

To be able to understand this behaviour a bit better, we selected a shape where the complexity can be assessed visually without ambiguity, unlike the talus or liver shapes. In this case, we considered a data set of 40 MRI foot images, where we segmented the calcaneus bone of the foot (Fig. 8.41). We selected the calcaneus shape because it has many shape features that will be accentuated when eroding the original shapes, creating more complex shapes, as shown in Fig. 8.42. The features will be smoothed when dilating the original shapes, generating less complex shapes, as seen in Fig. 8.42.



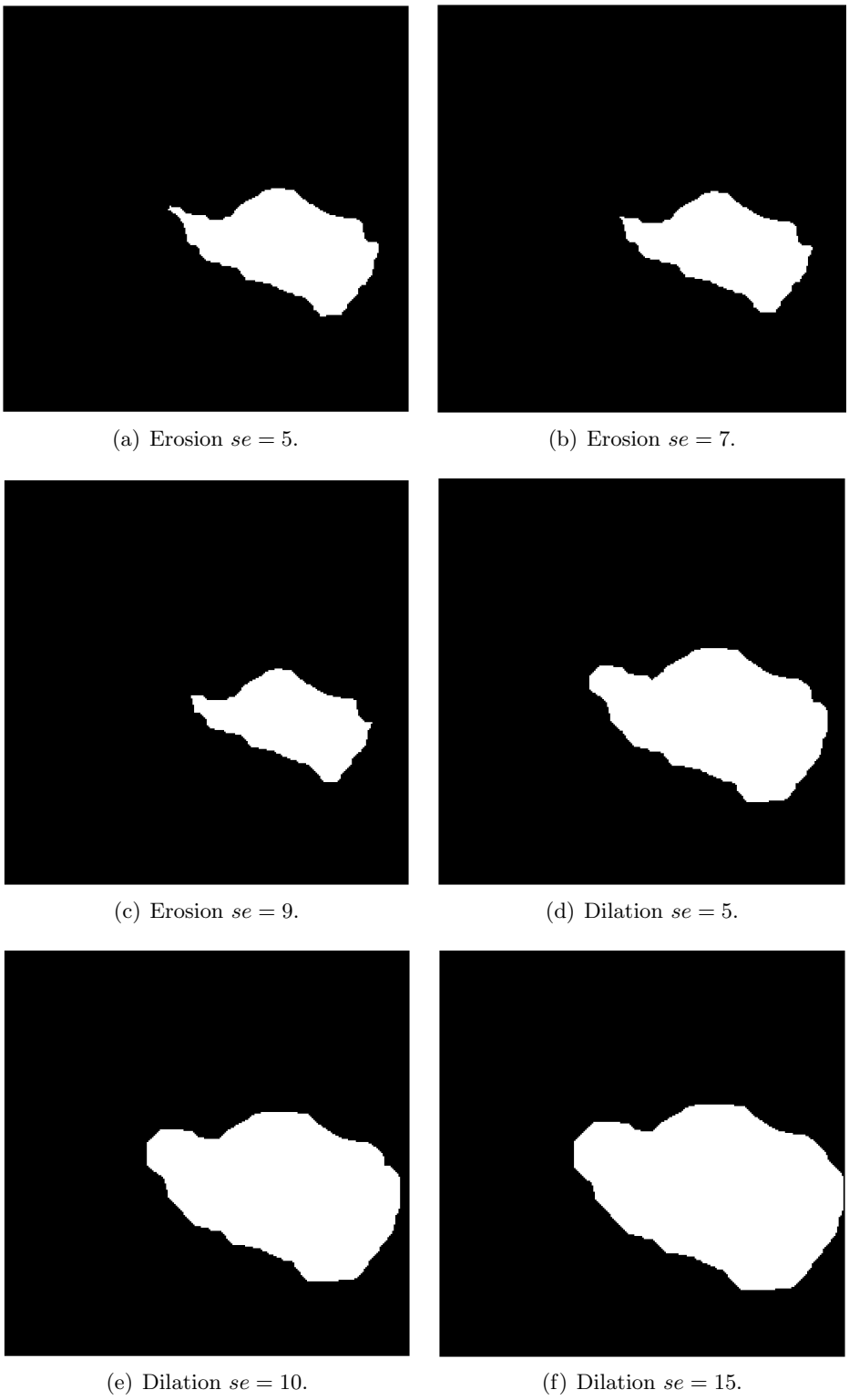
(a) MRI foot data. (b) Segmented calcaneus.

Figure 8.41: Example calcaneus bone taken from the training set of 40 shapes.

The $\chi(a)$ curves for these new shapes are displayed in Fig. 8.43. Note that the eroded training shapes start with a high compactness value for $a = 0$, which is the highest or similar to the compactness obtained at any point of these curves. For the dilated training sets, this is not the case, as the compactness, after reaching the minimum of the curve, increases and surpasses the value of χ at $a = 0$, reaching very high compactness values for large a. The position of the minimum varies as well with respect to the complexity of the shapes considered. The minima are deeper for the eroded families, and happen at a higher value of a than for the dilated families. The higher the erosion applied to the data set, the lower is the compactness curve obtained with respect to the parameter a. The higher the dilation applied to the data set, the higher the compactness curve and the faster it increases after reaching the minimum. This shows that there is some indication that the factor χ may depend on shape complexity. In the future, it would be interesting to explore this phenomenon fully.

8.5.4 Comparison of All Landmark Tagging Methods

To summarize the results obtained in this section, and to be able to compare among all methods in terms of compactness, we have represented the best results for each family of



(a) Erosion $se = 5$.

(b) Erosion $se = 7$.

(c) Erosion $se = 9$.

(d) Dilation $se = 5$.

(e) Dilation $se = 10$.

(f) Dilation $se = 15$.

Figure 8.42: Example of erosion and dilation for different sizes of *se* applied to the calcaneus shape in Fig. 8.41.(b).

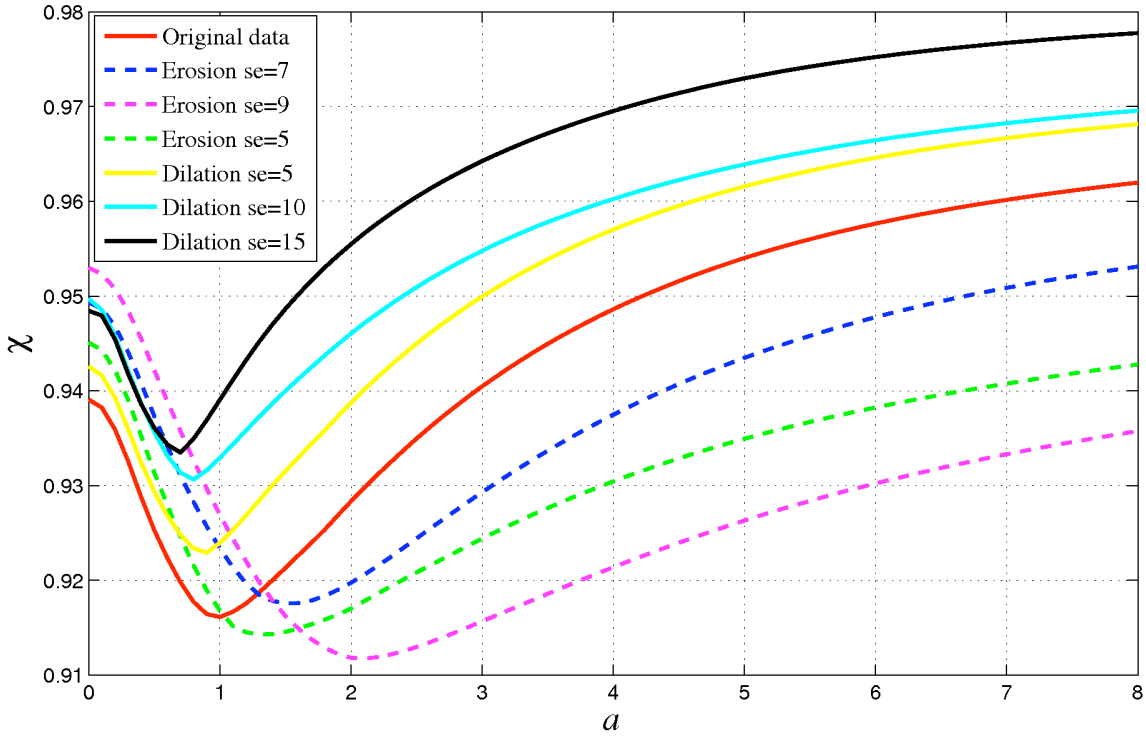


Figure 8.43: χ as a function of a. Selection of landmarks via virtual RBS based on eigenvalues for the calcaneus data set as shape complexity is changed.

methods for both training sets of shapes. For the talus bone, the comparison of all methods tested is presented, for $n = 1, ..., 18$, via χ_n in Fig. 8.44.

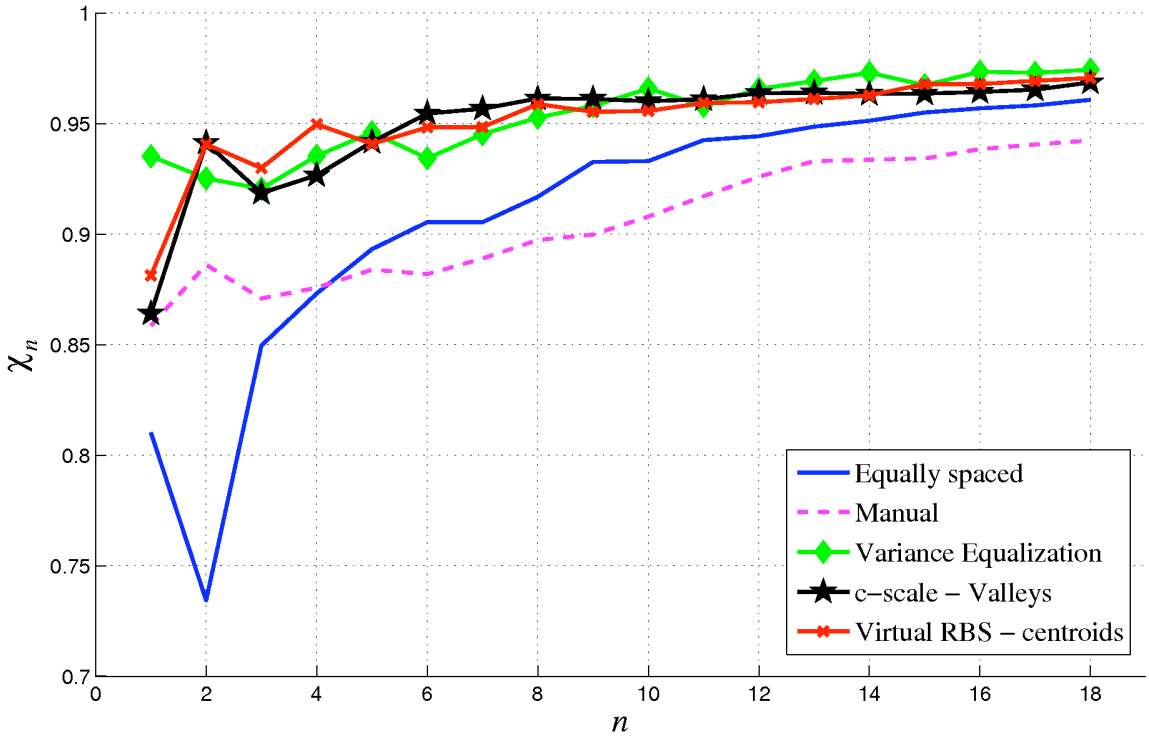


Figure 8.44: χ_n for $n = 1, ..., 18$. Selection of landmarks via c-scale, variance equalization, and RBS for the talus data set.

The best compactness for the c-scale approach was obtained when using the high curvature points (valleys) to describe the training shapes. For the RBS approach, the virtual strategy based on the centroids was the one with highest compactness. We observe that, among all the methods tested, the three families seem to perform similarly in terms on compactness for the talus training set with respect to χ_n. The curves corresponding to c-scale and virtual RBS seem to be quite similar, virtual RBS being a bit more compact for lower number of landmarks. All these methods perform better than manual and equally spaced annotation strategies. According to Table 8.11, in terms of χ, the values obtained for the three automatic landmark tagging methods are very similar, the best compactness being obtained by the variance equalization strategy, followed very closely by the virtual RBS approach. For the talus training set of shapes, c-scale, variance equalization, and RBS work significantly

Table 8.11: Comparison of the best variant of each method in terms of χ value for the talus data set.

	c-scale - Valleys	VE	Virtual RBS	Equally spaced	Manual
χ	0.95	0.9540	0.9516	0.9095	0.9065

better than manual and equally spaced approaches in terms of χ.

Similarly, for the liver training set, the comparison of all methods tested for $n = 1, ..., 15$ are displayed via χ_n in Fig. 8.45. The best compactness for the c-scale approach was obtained when using the inflection points (peaks). For the RBS approach, the PCA-based strategy was the one with highest compactness.

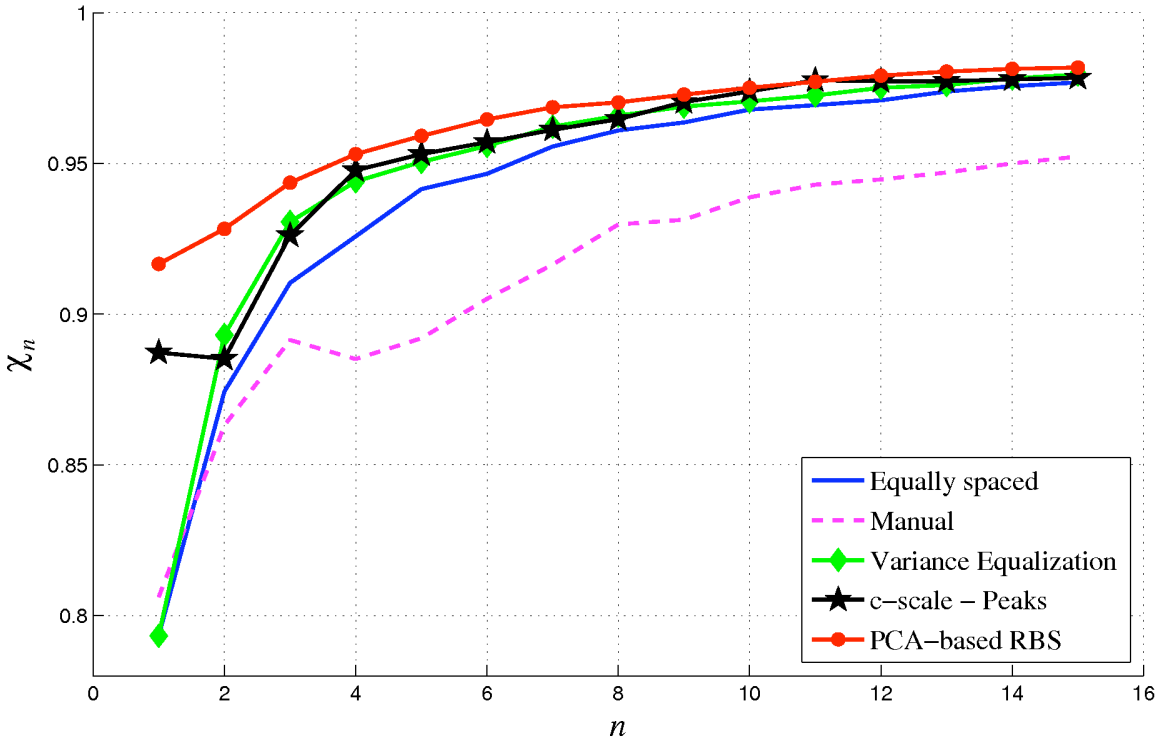


Figure 8.45: χ_n for $n = 1, ..., 15$. Selection of landmarks via c-scale, variance equalization, and RBS, for the liver data set.

From Fig. 8.45, we notice that the method with highest compactness is the PCA-based RBS. c-scale and variance equalization seem to perform similarly for all number of landmarks greater than 1. All three family of methods tested perform better than equally spaced and manual annotations. The same conclusions can be reached by examining at Table 8.12, where the results in terms of χ show that the RBS strategy is more compact than any other method tested, followed by the c-scale approach.

Table 8.12: Comparison of the best variant of each method in terms of χ value for the liver data set.

	c-scale - Peaks	**VE**	**PCA-RBS**	**Equally spaced**	**Manual**
χ	0.9544	0.9478	0.9635	0.9405	0.9131

8.6 Conclusions

In this chapter, we have compared the novel methods of automatic landmark tagging presented in this thesis. We evaluated the landmark selection achieved by each method and the compactness of the models considering both the number of landmarks and the modes selected and compared these methods with manual and equally spaced annotation methods. This analysis had a novel bent to it in the sense that it did not confine to some fixed number of landmarks n, arbitrarily determined, as it is commonly done in the literature, but it analysed how compactness varied with n. Our evaluation indicates that the three proposed families of new methods perform better than manual and equally spaced annotations. The reason why our methods perform better than manual, which is supposedly the ground truth, is that defining homology between landmarks manually is very difficult, time consuming, and error prone. Also, it is impossible to manually ascertain what specific landmarks may lead to the most compact model. Our analysis indicated that if n is made sufficiently large, all methods reach more or less the same compactness factor, although there may be some offset among the curves for different methods. Lower values of n show significant differences in behaviour among methods. If we had produced χ values by integrating up to smaller n

values, much larger differences will show in favour of the new methods. This underscores the point of showing the behaviour of χ_n over a larger range of n. It should be emphasized that, in the absence of reference standards and ground truth, even approaching χ for the manual method bodes well for a new method since that is the best intuitively agreeable tool we have for the selection of homologous points. The ability of automation is then the main selling attribute of the new methods, over manual selection.

c-scale approaches are able to capture dominant points at different scales (levels of detail) creating compact models for different number of landmarks. Since the shapes of a training set do not necessarily have the same shape characteristics, landmarks were selected directly on the mean shape of the training set and subsequently propagated to all the shapes of the training set establishing the correspondence between landmarks among shapes. Several strategies for selecting and propagating landmarks were evaluated in this chapter. When the number of landmarks needs to be varied repeatedly, the hierarchical approach of selection is to be preferred. Landmark propagation was undertaken by using two different approaches: closest point and parametric propagation. The results show that the latter returns more compact models in the two data sets analysed in this chapter. The type of landmark (peaks, valleys, or both) is to be selected depending on the application. Based on our analysis on a limited number of objects, we may draw the following conclusions. For objects with large flat segments, segment midpoints/ inflection points identified by peaks in $A_f(b)$ seem to be more appropriate as landmarks. For objects with more curved segments and details, valleys denoting high curvature points seem to capture shape information better. Generally hierarchical selection, which allows detecting points that are more dominant first then followed by subtler details, is to be preferred. Finally, parametric propagation is more suitable than closest point propagation since it locates the corresponding points along the boundary in a more consistent manner. Overall, we recommend hierarchical, parametric combinations on peaks or valleys or both.

We also showed that the distribution of landmarks by using the variance equalization

method is a good way of creating more compact models taking into account the variability existing in the training set. This concept has been missing in all previous attempts at automatic landmark tagging.

The recursive boundary subdivision strategy is a global-to-local approach for determining landmarks on digital boundaries. The method allows the incorporation of a variety of shape analysis methods on the sub-segments to determine landmarks. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at higher resolution. This method has a high potential, since different combinations of it can be useful for different applications. Especially the PCA-based method is very useful in characterizing a training set of shapes. This method has built into it the spirit of maintaining homology among all shapes starting with the most global and proceeding to as local a detail as we wish. We also introduced the new concept of virtual landmarks that relate to the shape but does not necessarily lie on the boundary of each shape.

These methods are in principle applicable to spaces of any dimensionality, although we have focused in this chapter on 2D shapes. Both local-to-global and global-to-local strategies have their merits and demerits. It remains to be seen how these will pan out especially in the 3D context.

9

Building 3D Shape Models Via Recursive Boundary Subdivision

In Chapter 7, we introduced the RBS method and detailed how to use it to build 2D models. In this chapter, we extend this approach to the 3D space, in an attempt to define and characterize landmarks on 3D shapes while ensuring their homology among shapes. The method starts from three initial corresponding points determined in all training shapes via a method α, and subsequently by subdividing the shapes into connected boundary segments by a plane determined by these points. A shape analysis method β is applied on each segment to determine a landmark on the segment. This point introduces more triplets of points, the planes defined by which in turn are used to further subdivide the boundary segments. This recursive boundary subdivision (RBS) process continues simultaneously on all training shapes, maintaining synchrony of the level of recursion, and thereby keeping correspondence among generated points automatically by the correspondence of the homologous shape segments in all training shapes. The process terminates when no subdividing planes are left to be considered that indicate, as per a parameter δ associated with method β, that a point can be continued to be selected on the associated segment. Similarly to the 2D approach, the selection of landmarks on 3D shapes is undertaken in a coarse-to-fine manner on discrete surfaces, with the same hierarchical strategy and associated properties as in the 2D approach.

In this chapter, we first describe the 3D RBS method, and we illustrate the approach with an example. Then, in Section 9.2, we delineate the overall algorithm of the method. Some

preliminary results are shown in Section 9.3 for a synthetic shape as well as for a training set of shapes constituting the talus bone of the foot of different human subjects. The conclusions are given in Section 9.4.

9.1 Principles of the 3D RBS Method

The geometric and topological properties of digital, oriented, connected, Jordan surfaces form an essential underpinning of the proposed RBS method. Let the coordinate axes of the space in which we represent each surface be denoted by x_1, x_2, and x_3. The boundary of any object is represented as a set of boundary faces, which are one of six types of oriented voxel faces, denoted $+x_1$, $+x_2$, $+x_3$, $-x_1$, $-x_2$, and $-x_3$ types. Each face in a surface is characterized by a center, four corners, and an orientation. The concept of boundary connectedness is essential to the understanding of the method. These concepts were already presented in Section 7.1 for 2D and 3D objects. Further details can be found in [Udupa, 1994].

Let $\mathcal{S} = \{S_j : 1 \leq j \leq M\}$ be the given set of training shapes represented as oriented, connected, and closed (Jordan) digital surfaces [Udupa, 1994], as defined in Chapter 7. The method can be generalized easily from a single Jordan surface per object to multiple Jordan surfaces and for multiple objects. We assume that the surfaces in $\mathcal{S}$ have been aligned via translation, scaling, and rotation as commonly done in most modelling methods. We will illustrate the approach for the two training shapes depicted in Fig. 9.1. We assume that both surfaces belong to the same family of shapes.

9.1.1 Initialization by Method α

Suppose by using some method α we find 3 initial homologous (non-collinear) points p_1^j, p_2^j, and p_3^j in each surface S_j in $\mathcal{S}$. This can be done by any method as long as it guarantees the correspondence between landmarks among shapes. In Fig. 9.2, we illustrate this step by defining α as the method that finds bels (voxel faces) with the largest distance between them in a shape. We initially find the two points p_1^j and p_2^j as the two farthest points on

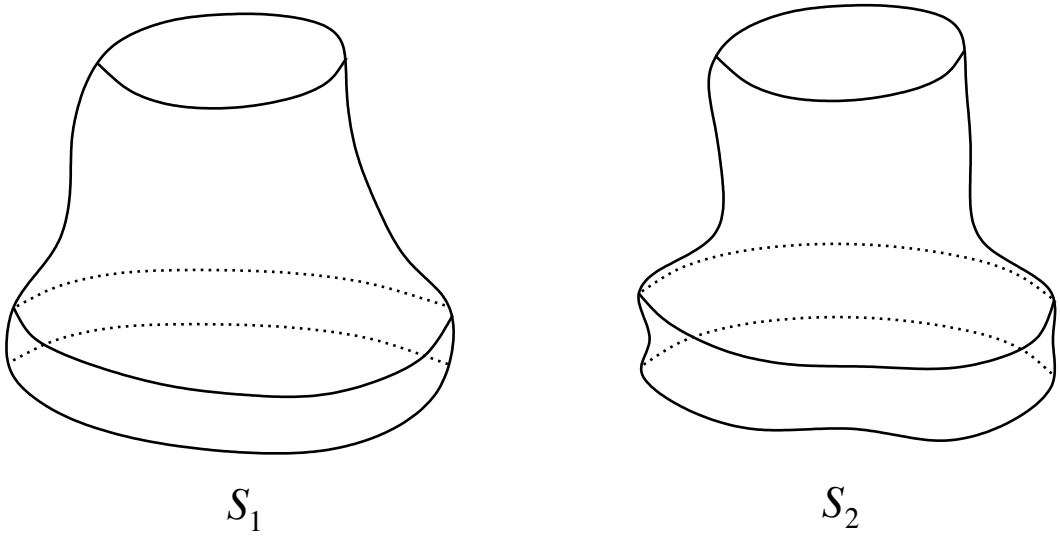


Figure 9.1: Training set of shapes S_1 and S_2.

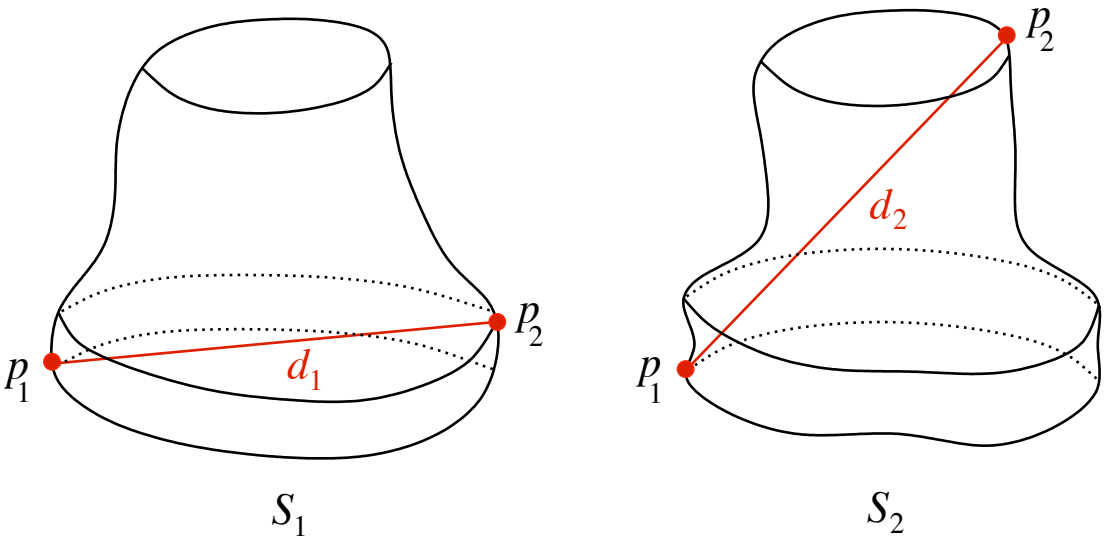


Figure 9.2: Defining landmarks p_1^j and p_2^j on S_j by method α.

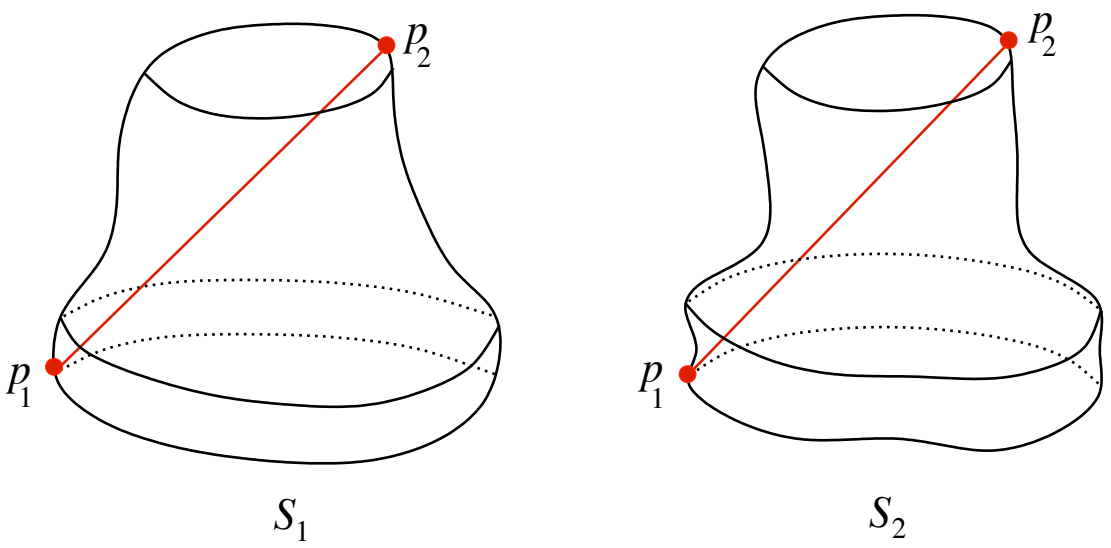


Figure 9.3: Propagation of landmarks p_1^2 and p_2^2 into S_1.

each surface S_j, as represented in Fig. 9.2. Then, we determine, among all S_j, the shape that has the largest distance between p_1^j and p_2^j. As for the 2D approach, there is only one shape leading each subdivision, and it is the shape that has the largest distance between p_1^j and p_2^j among shapes at that stage of subdivision. This is done to guarantee that landmarks are homologous among shapes at each step. Initially, the two shapes in our example may not necessarily have their farthest two points in equivalent positions. In Fig. 9.2, the largest distance happens for S_2. Therefore, p_1^2 and p_2^2 will be propagated to the surface S_1 to find the corresponding landmarks on that shape, as represented in Fig. 9.3. In this case, we used closest point propagation, as decribed in Section 5.5, to find the corresponding landmarks among shapes.

In the 2D case, we just needed two landmarks to initialize the method and divide a boundary shape into two connected pieces. In the 3D case, we need three landmarks to initialize the RBS approach. The third point p_3^j on S_j is determined to be the point farthest from the line $p_1^j p_2^j$ on S_j, as shown in Fig. 9.4. This is also done in a manner similar to p_1^j and p_2^j. In this case, $d_1 > d_2$, and therefore, point p_3^1 on S_1 will be propagated to all other surfaces to find all corresponding p_3^j, where $j \neq 1$. In this manner, by method α, we find the

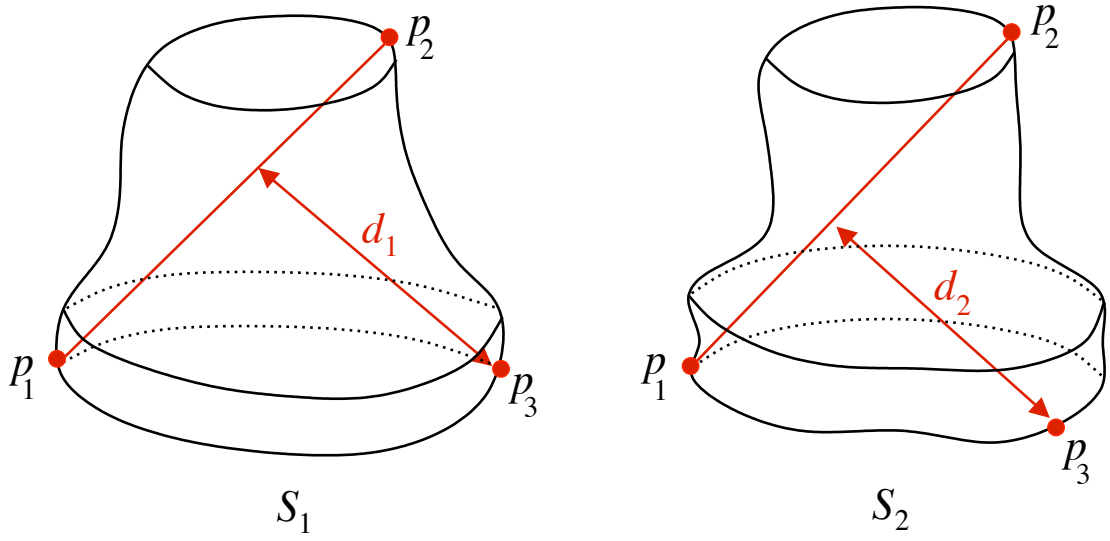


Figure 9.4: Defining landmark p_3^j with method α.

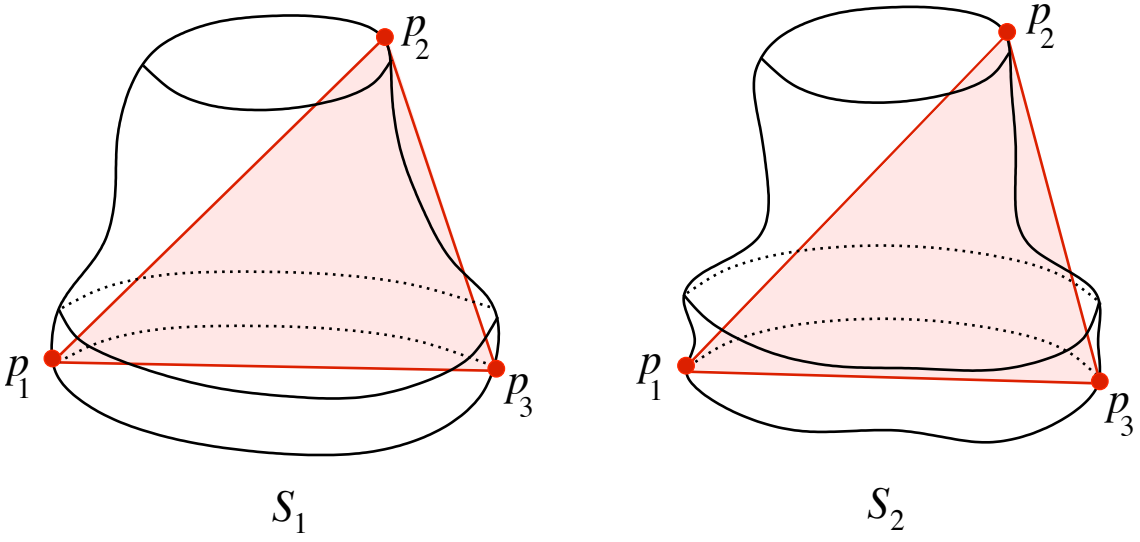


Figure 9.5: Propagating landmark p_3^1 into S_2.

triplet $t_1^j = \langle p_1^j, p_2^j, p_3^j \rangle$. The triplet t_1^j associated with S_j defines a plane $P(t_1^j)$ that intersects S_j at certain bels, as illustrated in Fig. 9.6 for the triplets shown in Fig. 9.5. The intersection between the planes and their corresponding surfaces is represented by a dotted blue line. To

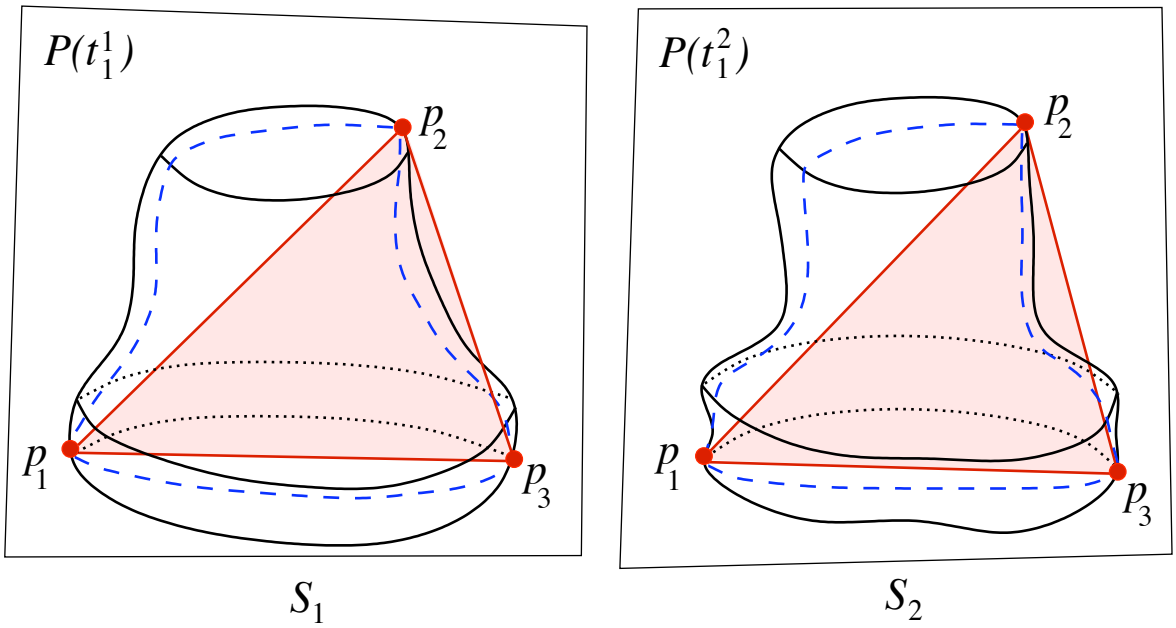


Figure 9.6: Intersecting faces (dotted blue line) between $P(t_1^j)$ and S_j.

subdivide S_j into two connected pieces, we find, from the set of intersecting faces those that are connected to p_1^j, p_2^j, and p_3^j. This will result in one or more connected sets of faces, which will lead to the further subdivision of each surface into two segments, connected to p_1^j, p_2^j, and p_3^j.

9.1.2 Determining The Intersecting Faces of a Surface by a Plane

For a given $P(t_1^j)$, the faces of a given type, say the x_1-faces, in S_j will be in one of the following situations in terms of their relationship with the plane:

1. They are parallel to $P(t_1^j)$;
2. One pair of their opposite (parallel) edges is parallel to $P(t_1^j)$;
3. No edge is parallel to $P(t_1^j)$.

In the first case, the face is entirely on the plane or out of it. In the second case, either the face intersects the plane or one of its edges touches it. In the last case, either the face intersects the plane or one of its vertices touches it. This is applicable to the x_2- and x_3-faces as well.

A simple way of finding the intersecting faces between $P(t_1^j)$ and S_j is as follows. The main idea is to use the center point of a face to decide if any part of the face is intersecting the plane, including the corners of the face in the case it touches the plane. This is possible due to the digital nature of the space. We create three generic faces of types x_1, x_2, and x_3, exactly touching the plane at a corner. If the faces belong to one of the first two cases enumerated above, then they will have more than a corner touching the plane. This does not affect the method, since the process of finding intersecting faces is the same independently of the relationship between the plane and the face considered. For the generic faces (as above) for each type of face, we calculate the distance of their center to the plane $P(t_1^j)$. These distances, denoted D_1, D_2, D_3, could be different for x_1-, x_2-, and x_3-faces because of their different orientation with respect to $P(t_1^j)$. Note that the orientation of all x_1-faces (and of $-x_1$-faces from the view point of determining intersection) is the same, as well as of x_2-, and x_3-faces. We determine an x_i-face (also $-x_i$-face) in S_j as intersecting $P(t_1^j)$ only if its centre's distance to the plane is less than D_i, for $i = 1, 2, 3$.

9.1.3 Finding Two Connected Sets

Once we find the intersecting faces of S_j with plane $P(t_1^j)$, we need to determine the set A_j among these faces that is connected to p_1^j, p_2^j, and p_3^j. A_j itself may not be connected in some cases. Let Q_1^j be the set of faces connected to p_1^j, Q_2^j the set connected to p_2^j, and Q_3^j the set connected to p_3^j, then, $A_j = Q_1^j \cup Q_2^j \cup Q_3^j$. The connectivity of bels is determined as explained in Section 7.1. By extracting the set of connected intersecting faces A_j from S_j, we divide each S_j into two segments, with the corresponding segments constituting roughly similar shape among all S_js, as shown in Fig. 9.7. Then, we find the connected sets on each side of the plane that are connected to a face adjacent to p_1^j, p_2^j, and p_3^j, separately, since we

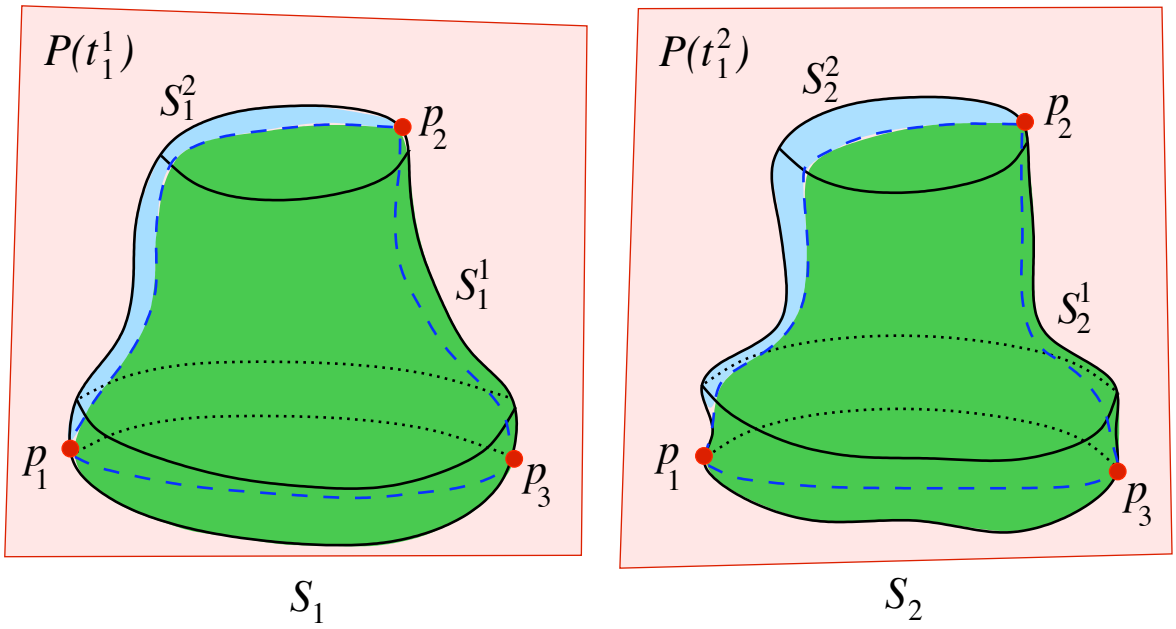


Figure 9.7: Subdivision of each S_j into two connected pieces S_j^1 and S_j^2.

can have several connected sets forming one segment. In Fig. 9.7, the planes $P(t_1^1)$ and $P(t_1^2)$ divide the surfaces S_1 and S_2 into two segments, formed by one connected piece each. The two corresponding segments among shapes are represented in green and blue. The dotted blue line is the cutting line of the plane $P(t_1^j)$. The two segments obtained for each S_j, after division by the plane, are denoted by S_j^1 and S_j^2.

9.1.4 Recursive Subdivision by Method β

Suppose now we use a method β, to find a point p_4^j in each new segment in S_j, guided by the shape of that segment. As in the previous explanation, we illustrate this step considering β as the method of finding the farthest point. In Fig. 9.8, p_4^j is the point that is the farthest from the plane $P(\langle p_1^j, p_2^j, p_3^j \rangle)$ for the segments S_j^1. Again, from both shapes considered in the training set, we take as final p_4^j, the one that has the largest distance to the plane among shapes. In this case, we select p_4^1 in this manner, it will lead the subdivision of all corresponding segments, and it will be propagated to S_2^1 to generate its corresponding p_4^2 (Fig. 9.9). A similar process will be undertaken to find a new landmark p_5^j on S_j^2.

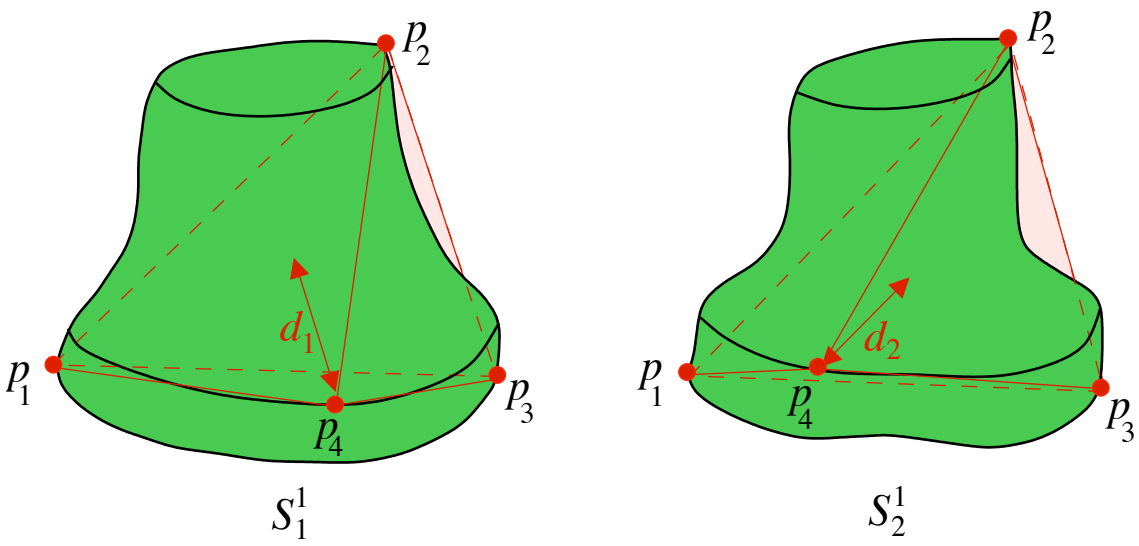


Figure 9.8: Finding a new landmark p_4^j on S_j^1 by method β.

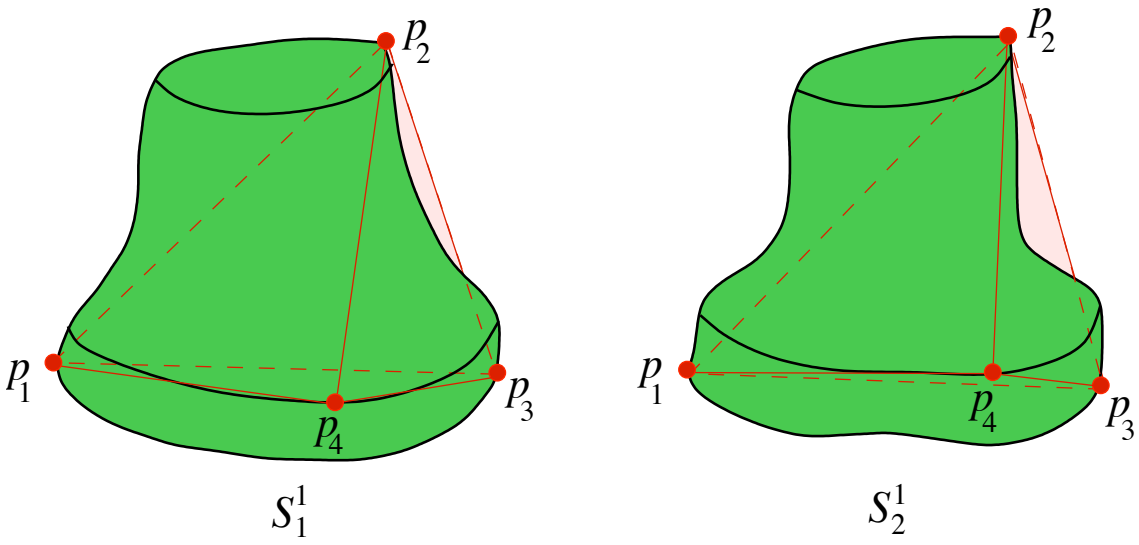


Figure 9.9: Propagating landmark p_4^1 to S_2^1.

The subdivision will cease if the surface segment under consideration, say S_j^1, does not deviate much from a planar surface as per parameter δ. Otherwise, three additional planes are introduced corresponding to the triplets $\langle p_1^j, p_2^j, p_4^j \rangle$, $\langle p_1^j, p_3^j, p_4^j \rangle$, $\langle p_2^j, p_3^j, p_4^j \rangle$, which further subdivide S_j^1 into three connected segments. The new planes generated for the segments S_j^1, are shown in Fig. 9.10 in cyan, magenta, and yellow, respectively.

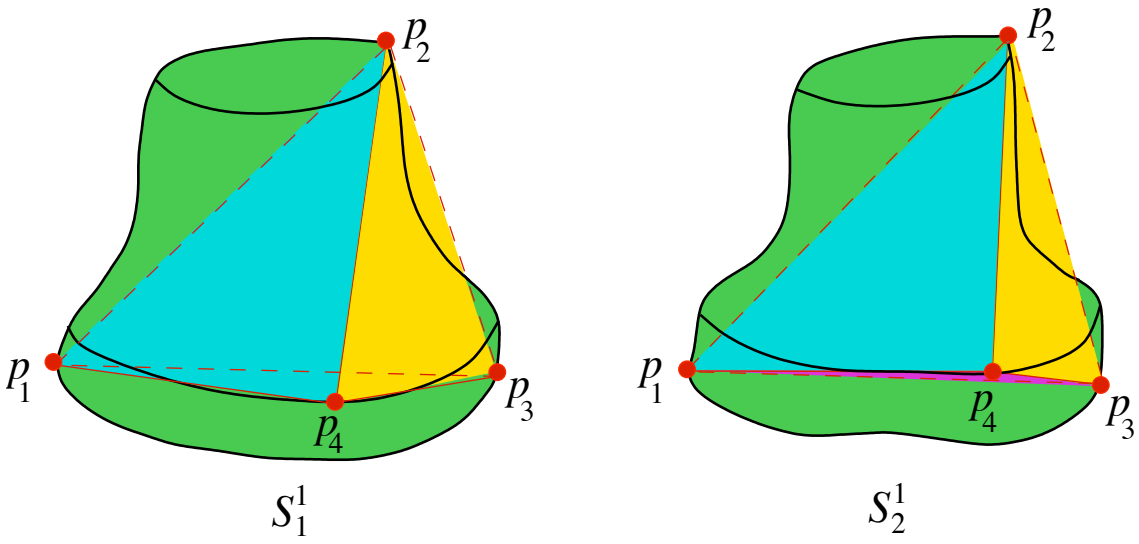


Figure 9.10: New planes for subdivision of S_j^1.

The process of subdivision continues until all planes and associated sub-segments indicate termination as per parameter δ. Note that the process of subdivision will continue if there is even one surface in the set $\mathcal{S}$ for which the distance criterion for method β is greater than the parameter δ. Otherwise, it will be terminated for that plane for all M surfaces. In the case of the example given, the subdivision will continue if there is any new point that is at a distance greater than δ from the plane that generated it.

9.2 Overall Algorithm

The algorithm used to implement the 3D RBS method is summarized below.

Algorithm 9.1: **3D RBS** - Recursive Boundary Subdivision

Input: Surfaces $S_j : j = 1, ..., M$; parameter δ; methods α and β.

Output: Landmarks $p_1^j, ..., p_n^j, j = 1, ..., M$.

Auxiliary Data: Queue Q_j, associated with each surface $S_j, j = 1, ..., M$.

Initialization

for $j = 1, ..., M$ **do**

1. Call method α to find the initial three landmarks p_1^j, p_2^j, p_3^j on S_j. These points define a plane $P(\langle p_1^j, p_2^j, p_3^j \rangle)$ that divides S_j into two connected segments S_j^1 and S_j^2. Each segment is connected to p_1^j, p_2^j, and p_3^j.

2. Enqueue the two surface segments S_j^X associated with S_j in Q_j. Note the triplet $\langle p_1^j, p_2^j, p_3^j \rangle$ associated with these segments.

end

Recursive Subdivision

while Q_j, $j = 1, ..., M$, *is not empty* **do**

3. Dequeue surface segment S_j^X from Q_j for each $j = 1, ..., M$. Using the associated triplet $\langle p_k^j, p_l^j, p_m^j \rangle$, determine if S_j^X needs to be further subdivided as per method β and test criterion using δ.

4. If there is some i in $1, ..., M$ such that S_i^X can be subdivided, then, for each $j = 1, ..., M$, determine the new segment of S_j^X using the triplet $\langle p_k^j, p_l^j, p_m^j \rangle$, and enqueue the segment in Q_j.

end

5. Output $p_1^j, ..., p_n^j, j = 1, ..., M$.

9.3 Preliminary Demonstration

In this section, we present some preliminary results to mainly demonstrate the spirit of the algorithm with some examples. We have defined methods α and β as being the farthest distance and farthest distance from the plane to the surface, respectively. We applied the 3D RBS method to two 3D objects: a synthetic shape with easily identifiable corners as shown in Fig. 9.12, and a medical object - the talus bone of the foot, as represented in Fig. 9.11.

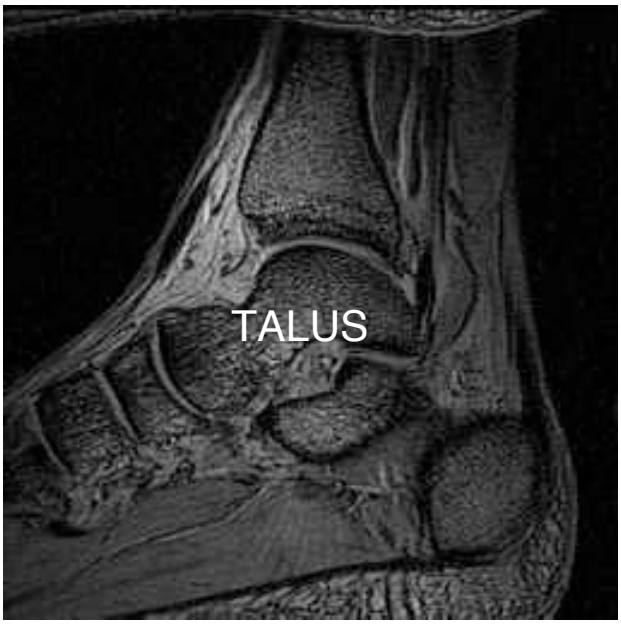


Figure 9.11: The talus bone of the foot in an MR image.

In Fig. 9.12, we show the results obtained after applying the 3D RBS method to a single synthetic shape. The idea here was to show how landmarks are well positionned by the RBS method in the corners of the shape. Different views of the same surface are displayed in Fig. 9.12. Landmarks are represented as small balls attached to the surface.

For the talus object, the process of localizing landmarks is illustrated in Figs. 9.13-9.15. The initial three landmarks for two of the training shapes are displayed in Fig. 9.13. In this figure, we show similar views for each shape. We notice that the initial landmarks were selected on the surface at corresponding positions. Furthermore, according to the selected δ

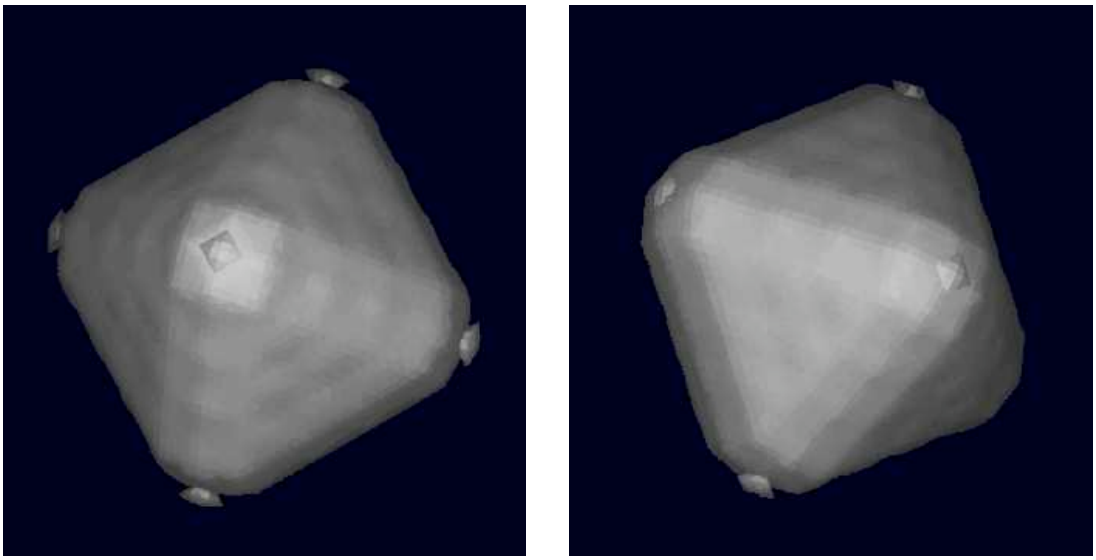


Figure 9.12: A synthetic shape with its localized landmarks.

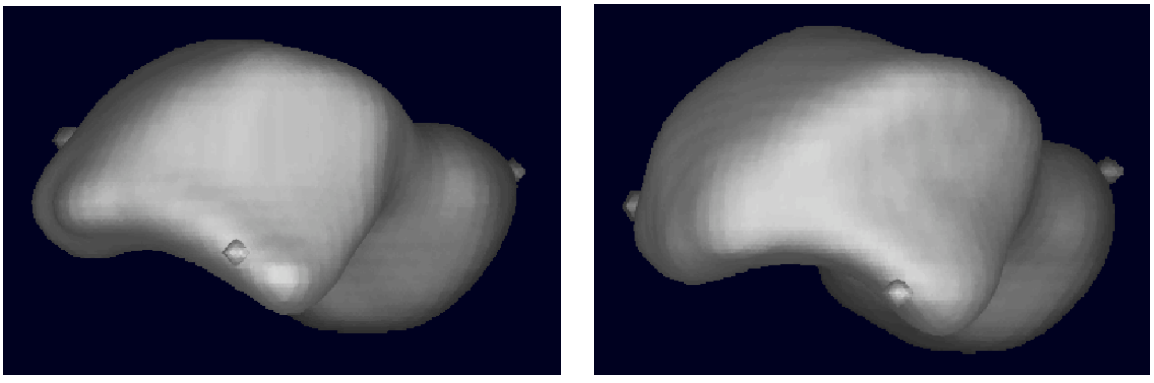


Figure 9.13: Initialization of the RBS method on two tali.

parameter, we can capture different levels of detail on the surface as shown in Figs. 9.14 and 9.15, where two different values of delta were selected.

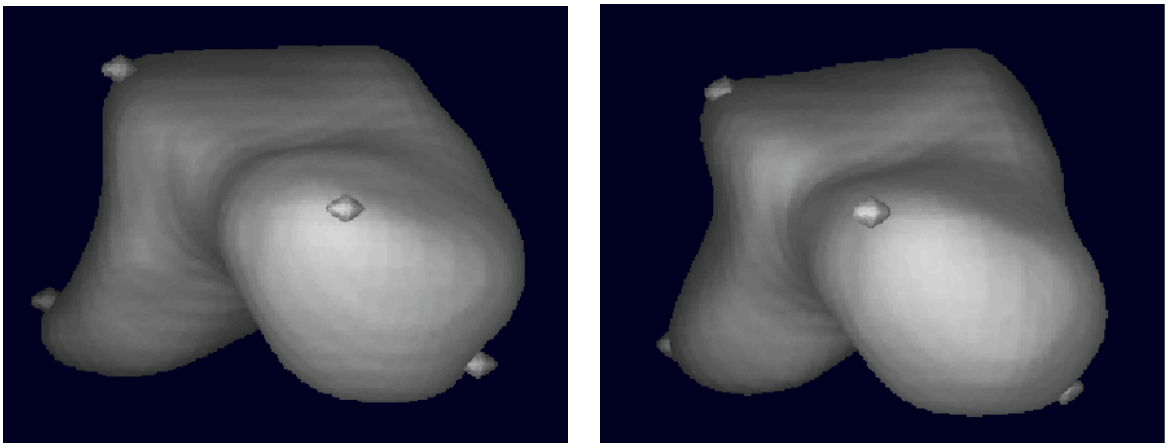


Figure 9.14: Landmarks obtained on two tali by using the RBS method with a higher δ value.

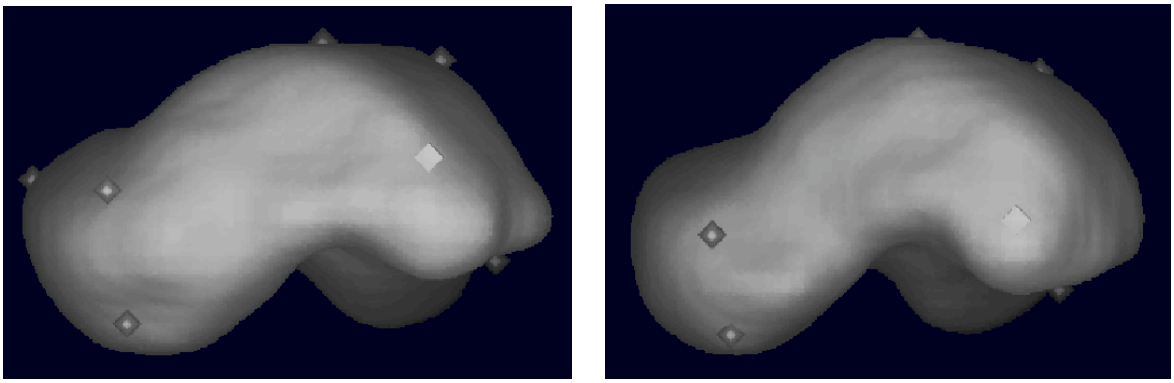


Figure 9.15: Landmarks obtained on two tali by using the RBS method with a lower δ value.

The results are promising and show how well landmarks are captured on different types of surfaces and with different levels of detail and at corresponding locations. If the method is applied to a single shape, as shown in Fig. 9.12 for the synthetic shape, it is able to capture the dominant points of the shape, acting as a shape descriptor to the surface.

9.4 Conclusion

In this chapter, we have presented a 3D automatic landmark tagging method based on global-to-local recursive boundary subdivision for determining landmarks on $3D$ digital surfaces. The method is an extension of the 2D approach presented in Chapter 7 and presents the following characteristics. (a) The method is general in being nD and in allowing the incorporation of a variety of shape analysis methods on the sub-segments to determine landmarks. (b) Its output is naturally multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of the landmarks found at higher resolutions. (c) The operations are very efficient requiring minutes rather than many hours (for the $3D$ case) for model building. (d) A triangulation of the shape surface (at all levels of resolution) is a byproduct of the method which becomes useful in visualization, segmentation, and other operations involving the model. In fact, this is a very interesting surface triangulation method based on global-to-local shape refinement.

In this chapter, we have demonstrated that the idea underlying RBS can be extended to the 3D space. However, this is only a preliminary demonstration. Much further work is needed in exploring the full potential of the method, in developing a production mode robust implementation, and in comprehensively evaluating the method on a large population of medical objects.

10

Concluding Remarks

The goal of the thesis was to address challenges C1-C4 in the best way it was possible. Instead of jumping right into the 3D problem first, it seemed reasonable to look at the 2D problem first since 2D modelling itself has numerous applications. This naturally led us to the c-scale idea. Although the c-scale concept brought on several new ideas, which led to considerable improvements in salient point detection from the state-of-the-art, since the same dominant points may not occur in all training shapes, it posed a hurdle for the c-scale based landmark tagging strategies. This led us to the variance equalization strategy for 2D landmark selection. The RBS strategy's strongest tenet is its n-dimensional generalizability, which is a lot more challenging for the other two approaches studied in the thesis. Unfortunately time restrictions precluded a full and thorough study of the RBS methodology.

This chapter summarizes the conclusions of this thesis. Section 10.1 states the main contributions achieved by this work. A discussion is given in Section 10.2. In Section 10.3, we present some open problems that remain to be solved in the future.

10.1 Main Contributions

The main contributions of this thesis are summarized below under the different topics tackled in this thesis.

Shape Description

- We have devised a new theory and method for shape description based on the novel concept of c-scale. This shape descriptor provides a comprehensive characterization of shape with numerous potential applications. From this representation, we can extract special points of interest such as convex and concave corners, straight lines, circular segments, and inflection points. The method is simple, different from previous methods of curvature estimation, and can be directly applied to digital boundaries without requiring prior approximations of the boundary, giving robust and accurate results at different levels of detail by considering the local morphometric scale of the object. We have shown that this method is useful for shape description as well as for the extraction of dominant points.

Automatic Landmark Tagging Methods

- We devised a novel automatic landmark tagging method based on c-scale. The c-scale method is applied to the mean shape of a training set of shapes to extract dominant points such as high curvature points, inflection points, or both. These approaches based on local curvature are able to capture dominant points at different scales (levels of detail) and then propagate them to all the shapes of the training set to establish correspondence in a local-to-global manner. Several techniques for landmark selection and landmark propagation were evaluated.

- We developed a novel strategy for automatic landmark tagging that uses a technique of equalization of the total variance existing in the training set, automatically establishing correspondence among shapes at the same time. The main premise of this approach is that regions with high variance among shapes will need more landmarks to be described consistently, whereas regions with low variance do not need to be described with many landmarks. This strategy itself takes care of the correspondence issue, and at the same time, deploys landmarks very frugally and optimally considering shape variations. The desired landmarks are positioned around each shape boundary so as to equally

distribute the total variance existing in the training set in a global-to-local manner. This is a consideration that has been missing in all landmark tagging methods proposed up to now. The variance equalization method is simple to implement, computationally fast, and does not need a reference shape to establish landmark correspondence as many other methods in the literature.

- We devised a novel automatic landmark tagging method based on a recursive boundary subdivision process. This is a global-to-local strategy that directly addresses the generalization of methods to $\mathfrak{n} > 2$ dimensions, while making computations practical, and operating in $\Re^{\mathfrak{n}}$. It attempts to mathematically define and characterize landmarks and ensures the homology among landmarks via recursive boundary subdivision (RBS) of all training shapes simultaneously, maintaining correspondence among shapes and their subdivided segments, as per a specified shape similarity criterion, at every recursive step. The method is computationally simple and does not require registration but uses the fundamental attributes of the shapes in the training set. It undertakes the selection of landmarks in a coarse-to-fine manner, consulting all objects in the training set simultaneously in establishing homology and in finding landmarks. The method allows the incorporation of a variety of shape analysis methods on the sub-segments to determine landmarks. Its output is multi-resolutional allowing landmark selection at any lower resolution trivially as a subset of those found at a higher resolution. We presented the method in the 2D space and extended it to the 3D case, displaying some preliminary results in the latter case.

- Another novelty of this thesis is the introduction of the new concept of virtual landmarks that relate to the shape but does not necessarily lie on the boundary of each shape. This has been applied in the context of RBS strategies, but it could be used in any other automatic landmark tagging method.

Evaluation

- We introduced several new compactness factors to assess the performance of the methods independently of the number of modes and the number of landmarks considered. Our study did not confine to some fixed number of landmarks n, arbitrarily determined, as it is commonly done in the literature, but it analysed how compactness varied with n.

10.2 Discussion

Shape Description

We compared global and local scale approaches to shape analysis and presented a thorough evaluation on 21 different kinds of natural and artificial shapes. Based on accuracy in detection (FPs and FNs) and localization accuracy (% RMS distance), c-scale method was the best among all state-of-the-art and common methods used for comparison. The method was also capable of detecting inflection points with good accuracy of detection and localization, unlike many of the methods in the literature. It was the only method with perfect detection accuracy and which was consistent across all shapes, demonstrating non dependence on shape in performance, even for features that are small or large in size and sharp or subtle in definition. Furthermore, the c-scale method can control the level of detail and is robust to scale changes. In that sense, it is also better than the other methods tested. The c-scale method is an adaptation of the ball-scale idea [Saha et al., 2000] for nD images to the 1D boundaries of 2D shapes. It produces a richer description of shape than the other methods compared. It is simpler, with no special purpose mechanisms, has a simple implementation, clear definition, and has a lower computational cost than any of the other methods used for comparison purposes.

Automatic Landmark Tagging

For the c-scale family, we devised two different landmark selection strategies - hierarchical and non-hierarchical. Landmark propagation was undertaken by using two different approaches: closest point and parametric propagation. When the number of landmarks needs to be varied repeatedly, the hierarchical approach of selection is to be preferred. The result of the comparison shows that parametric propagation returns more compact models in the two data sets analysed. The type of landmark (peaks, valleys, or both) is to be selected depending on the application. For objects with large flat segments, segment midpoints/inflection points identified by peaks seem to be more appropriate as landmarks. For objects with more curved segments and details, valleys denoting high curvature points seem to capture shape information better. Generally hierarchical selection, which allows detecting points that are more dominant first then followed by subtler details, is to be preferred. Finally, parametric propagation is more suitable than closest point propagation since it locates the corresponding points along the boundary in a more consistent manner. Overall, we recommend hierarchical, parametric combinations on peaks or valleys or both. Other more advanced techniques could be also used as propagation method for this family of methods.

In the variance equalization approach, the landmarks may not correspond to any dominant points on the shape as in previous strategies, but they will give a good representation of the variability existing in the training set. This is an important idea that has been missing in previous methods and that allows a model to incorporate all the variability information existing in the training set of shapes, so that the variability can be characterized directly and in a better manner. These methods seem to perform very well in terms of compactness for both objects studied.

The recursive boundary subdivision strategy is a global-to-local approach for determining landmarks on digital boundaries. This RBS process continues simultaneously on all training shapes, maintaining synchrony of the level of recursion, and thereby keeping correspondence

among generated points automatically by the correspondence of the homologous shape segments in all training shapes. This method has, we believe, much potential, since different combinations of it can be useful for different applications. Especially the PCA-based method is very useful in characterizing a training set of shapes. This method has built into it the spirit of maintaining homology among all shapes starting with the most global and proceeding to as local a detail as we wish. We have also demonstrated that the method can be extended to the 3D space by showing some preliminary results.

The evaluation results indicate that the three proposed families of new methods perform better than manual and equally spaced annotations for both objects studied. Our analysis shows that, if the number n of landmarks is made sufficiently large, all methods reach more or less the same compactness factor. Lower values of n show significant differences in behaviour among methods. If we had produced χ values by integrating up to smaller n values, much larger differences will show in favour of the new methods. This underscores the point of showing the behaviour of χ_n over a larger range of n. It should be emphasized that, in the absence of reference standards and ground truth, even approaching χ for the manual method bodes well for a new method since that is the best intuitively agreeable tool we have for the selection of homologous points. The ability of automation is then the main selling attribute of the new methods, over manual selection. Both local-to-global and global-to-local strategies have their merits and demerits. It remains to be seen how these will pan out especially in the 3D context in the future.

10.3 Some Open Problems

A more complete evaluation could be performed on the c-scale shape descriptor to study its behavior under noise and transformations and compare it to the state-of-the-art shape descriptors.

The future work consists of obtaining a full production level implementation of 3D auto-

matic landmark tagging approaches. In this thesis, we mainly focused on 2D shapes. This itself has many medical (radiographic projection images) and non medical applications. Although it is conceivable that the c-scale concept can be generalized to 3D shapes by using geodesic distances on surfaces, the extension is not straightforward. Therefore, it remains to be studied as to how to extend c-scale based and variance based landmark tagging strategies into the 3D space. Furthermore, even though if we have demonstrated in this thesis that the idea underlying RBS can be extended to the 3D space, this is only a preliminary demonstration. Much further work is needed in exploring the full potential of the method, in developing a production mode robust implementation, and in comprehensively evaluating the method on a large population of medical objects.

The evaluation undertaken was mainly in terms of compactness and landmark location. In the future, other evaluation factors could be studied to see how generalizable or specific the models created are, where again the number l of modes selected and n become independent variables. Furthermore, it would be interesting to see how well the different models perform when applied to model-based segmentation strategies in identifying the object of interest in new images of the same family.

In the future, we will complete this work to automate the whole process of generating the PDM based on the automatic landmark tagging methods.

Finally, some types of landmarks may not be derivable from purely geometric shape considerations only. There may be other indicators such as intensity patterns that signal "points of interest". The methods described in this thesis may be applicable to such situations only if the intensity pattern information can be converted into geometric information suitable, for example, in c-scale analysis.

Bibliography

[Andaló et al., 2007] Andaló, F.A., Miranda, P.A.V., Torres, R. da S., and Falcão, A.X., 2007. Detecting contour saliences using tensor scale. IEEE International Conference on Image Processing (ICIP). 6, 349-352.

[Asada and Brady, 1986] Asada, H. and Brady, M., 1986. The curvature primal sketch. IEEE Transactions on Pattern Analysis and Machine Intelligence. 8(1), 2-14.

[Attneave, 1954] Attneave, F., 1954. Some informational aspects of visual perception, Psychological Review. 61(3), 183-193.

[Baumberg and Hogg, 1994] Baumberg, A. and Hogg, D., 1994. Learning Flexible Models from Image Sequences. European Conference on Computer Vision, Springer-Verlag. 299-308.

[Berreti et al., 2000] Berretti, S., Bimbo, A. D., and Pala, P., 2000. Retrieval by shape similarity with perceptual distance and effective indexing. IEEE Transactions on Multimedia. 2(4), 225-239.

[Blum, 1967] Blum, H., 1967. A transformation for extracting new descriptors of shape. In Models for the Perception of Speech and Visual Forms, MIT Press, Cambridge, W. Whaten-Dunn (Ed.). 362-380.

[Bookstein, 1997] Bookstein, F.L., 1997. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. Medical Image Analysis, Oxford University Press. 1(3), 225-243.

[Brett and Taylor, 2000] Brett, A.D. and Taylor, C.J., 2000. A method of automated landmark generation for automated 3D PDM construction. Image and Vision Computing, Elsevier Inc. 18(9), 739-748.

[Breu et al., 1995] Breu, H., Gil, J., Kirkpatrick, D. and Werman, M., 1995. Linear Time Euclidean Distance Transform Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence. 17(5), 529-533.

[Carmona et al., 2005] Carmona-Poyato, A., Fernandez-Garcia, N.L., Medina-Carnicer, R., and Madrid-Cuevas, F.J., 2005. Dominant Point Detection: a new proposal. Image and Vision Computing. 23(13), 1226-1236.

[Chellapa and Bagdazian, 1984] Chellapa, R. and Bagdazian, R., 1984. Fourier coding of image boundaries. IEEE Transactions on Pattern Analysis and Machine Intelligence. 6(1), 102-105.

[Christensen et al., 1997] Christensen, G. E., Joshi, S. C., and Miller, M. I., 1997. Volumetric transformation of brain anatomy. IEEE Transactions on Medical Imaging. 16(6), 864-877.

[Cootes et al., 2001] Cootes, T. F, Edwards, G. J., and Taylor, C.J., 2001. Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 23(6), 681-685.

[Cootes et al., 1994] Cootes, T. F, Hill, A., Taylor, C.J., and Haslam, J., 1994. The Use of Active Shape Models for Locating Structures in Medical Images. Image and Vision Computing, Elsevier. 12(6), 355-366.

[Cootes et al., 1995] Cootes, T. F, Taylor, C. J, Cooper, D. H, and Graham, J, 1995. Active Shape Models - their training and application. Computer Vision Image Understanding, Elsevier Science Inc., New York, NY, USA. 61(1), 38-59.

[Cootes et al., 2008] Cootes, T.F., Twining, C., Babaola, K., and Taylor, C. J., 2008. Diffeomorphic statistical shape models. Image and Vision Computing. 26, 326-332.

[Costa and Cesar Jr., 2001] Costa, L. da F., and Cesar Jr., R. M., 2001. Shape Analysis and Classification: Theory and Practice, CRC Press.

[Davis, 1977] Davis, L. S., 1977. Understanding Shape: Angles and Sides. IEEE Transactions on Computers. 26(3), 236-242.

[Davies, 1997] Davies, E. R., 1997. Machine Vision: Theory, Algorithms, Practicabilities. Academic Press.

[Davies et al., 2002] Davies, H.D., Twining, C.J., Cootes, T.F, Waterton, J.C., and Taylor, C.J., 2002. A Minimum Description Length Approach to Statistical Shape Modeling. IEEE Transactions on Medical Imaging. 21(5), 525-537.

[Davies et al., 2003] Davies, H.D., Twining, C.J., Allen, P.D., Cootes, T.F, and Taylor, C.J., 2003. Building optimal 2D statistical shape models. Image and Vision Computing, Elsevier Inc. 21(13-14), 1171-1182.

[Dryden and Mardia, 1998] Dryden, I. and Mardia, K.V., 1998. Statistical Shape Analysis. John Wiley and Sons Ltd., Chichester, West Sussex, England.

[Falcão et al., 2000] Falcão, A.X., Udupa, J.K., Samarasekera, S., Sharma, S., Hirsch, B., and Lotufo, R., 2000. User-steered image segmentation paradigms: Live wire and live lane. Graphical Models and Image Processing, Academic Press, Inc., Orlando, FL, USA. 60(4), 233-260.

[Falcão et al., 2004] Falcão, A.X., Stolfi, J., Lotufo, R.A., 2004. The image foresting transform: theory, algorithms, and applications. IEEE Transactions of Pattern Analysis and Machine Intelligence. 26(1), 19-29.

[Frangi et al., 2001] Frangi, A., Rueckert, D., Schnabel, J.A., and Niessen, W.J., 2001. Automatic 3D ASM construction via atlas-based landmarking and volumetric elastic registra-

tion. Conference on Information Processing in Medical Imaging, Springer-Verlag, London, UK. 2082, 78-91.

[Freeman, 1961] Freeman, H., 1961. On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers. 10(2), 260-268.

[Freeman, 1978] Freeman, H., 1978. Shape description via the use of critical points. Pattern Recognition. 10(3), 159-166.

[Freeman and Saghri, 1980] Freeman, H. and Saghri, J., 1980. Comparative analysis of line-drawing modeling schemes. Computer Graphics and Image Processing. 12(3), 203-223.

[Fu, 1982] Fu, K.S., 1982. Syntactic Pattern Recognition and Applications. Prentice Hall, Englewood Cliffs, New Jersey, USA.

[Gee et al., 1993] Gee, J. C., Reivich, M., and Bajcsy, R., 1993. Elastically deforming 3D atlas to match anatomical brain images. Journal of Computer Assisted Tomography. 17(2), 225-236.

[Grosky and Mehrotra, 1990] Groskey, W. I. and Mehrotra, R., 1990. Index-based object recognition in pictorial data management. Computer Graphics and Image Processing. 52(3), 416-436.

[Grosky et al., 1992] Groskey, W. I., Neo, P., and Mehrotra, R., 1992. A pictorial index mechanism for model-based matching. Data Knowledge Engineering. 8(4), 309-327.

[Hermann, 2007] Hermann, S. and Klette, R., 2007. A comparative study on 2D curvature estimators. Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07). 584-589.

[Hill et al., 1997] Hill, A., Brett, A.D., and Taylor, C.J., 1997. Automatic landmark identification using a new method of non-rigid correspondence. Conference on Information Processing in Medical Imaging, Springer-Verlag, Berlin, Germany. LNCS 1230, 483-488.

[Klette and Rosenfeld, 2004] Klette, R., and Rosenfeld, A., 2004. Digital Geometry: Geometric Methods for Digital Picture Analysis. Morgan Kaufmann Publishers, Elsevier, San Francisco, USA.

[Kendall, 1977] Kendall, D.G., 1977. The diffusion of shape. Advances in Applied Probability. 9, 428-430.

[Li, 1999] Li, S. Z., 1999. Shape matching based on invariants. In Omidvar O (Ed.), Shape Analysis, Progress in Neural Networks. 6, 203-228.

[Lindeberg, 1994] Lindeberg, T., 1994. Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, Dordrecht, Netherlands.

[Loncaric, 1998] Loncaric, S., 1998. A survey of shape analysis techniques. Pattern recognition. 38(8), 983-1001.

[Lu and Sajjanhar, 1999] Lu, G.J. and Sajjanhar, A., 1999. Region-based shape representation and similarity measure suitable for content-based image retrieval. Multimedia Systems. 7(2), 165-174.

[Madabhushi et al., 2006] Madabhushi, A., Udupa, J.K., and Souza, A., 2006. Generalized Scale: Theory, Algorithms, and Application to image inhomogeneity correction. Computer Vision and Image Understanding, Elsevier Inc. 101(2), 100-121.

[Marsland and Twining, 2003] Marsland, S., and Twining, C., 2003. Constructing data-driven optimal representations for iterative pairwise non-rigid registration. Biomedical Image Registration, LNCS. 50-60.

[Miranda et al., 2005] Miranda, P. A. V., Torres, R. da S., and Falcão, A. X., 2005. TSD: a shape descriptor based on a distribution of tensor scale local orientation. Proceedings of Brazilian Symposium on Computer Graphics and Image Processing. 139-146.

[Mokhtarian et al., 2003] Mokhtarian, F. and Bober, M., 2003. Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization. Series: Computational Imaging and Vision, Kluwer Academic Publishers, Dordrecht, Netherlands. 25.

[Mokhtarian and Mackworth, 1986] Mokhtarian, F. and Mackworth, A., 1986. Scale-based description and recognition of planar curves and two dimensional shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 8(1), 34-43.

[Mokhtarian and Mackworth, 1992] Mokhtarian, F. and Mackworth, A., 1992. A theory of multiscale, curvature-based shape representation for planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence. 14(8), 789-805.

[Morse, 1994] Morse, B.S., 1994. Computation of object cores from grey-level images. PhD. Thesis, University of North Carolina at Chapel Hill.

[Neumann and Teisseron, 2002] Neumann, R. and Teisseron, G., 2002. Extraction of dominant points by estimation of the contour fluctuations. Pattern Recognition. 35(7), 1447-1462.

[Periaswamy, 2003] Periaswamy, S., 2003. General-purpose medical image registration. Ph.D. Dissertation, Department of Computer Science, Dartmouth College.

[Periaswamy and Farid, 2003] Periaswamy, S. and Farid, H., 2003. Elastic Registration in the Presence of Intensity Variations. IEEE Transactions on Medical Imaging. 22(7), 865-874.

[Periaswamy and Farid, 2006] Periaswamy, S. and Farid, H., 2006. Medical Image Registration with Partial Data. Medical Image Analysis. 10, 452-464.

[Pizer et al., 2003a] Pizer, S.M., Gerig, G., Joshi, S.C., and Aylward, S.R., 2003. Multiscale medial shape-based analysis of image objects. Proceedings of the IEEE. 91(10), 1670-1679.

[Pizer et al., 2003b] Pizer, S.M., Fletcher, P.T., Joshi, S.C., Thall, A., Chen, J.Z., Fridman, Y., Fritsch, D.S., Gash, A.G., Glotzer, J.M., Jiroutek, M.R., Lu, C., Muller, K.E., Tracton, G., Yushkevich, P.A., and Chaney, E.L., 2003. Deformable m-reps for 3D medical image segmentation. International Journal of Computer Vision. 55(2-3), 95-106.

[Ramer, 1972] Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing. 1(3), 244-256.

[Rattarangsi and Chin, 1992] Rattarangsi, A. and Chin, R.T., 1992. Scale-Based Detection of Corners of Planar Curves. IEEE Transactions on Pattern Analysis and Machine Intelligence. 14 (4), 430-449.

[Ray and Pandyan, 2003] Ray, B.K. and Pandyan, R., 2003. ACORD: an adaptive corner detector for planar curves. Pattern Recognition. 36(3), 703-708.

[Ray, B.K and Ray, K.S, 1995] Ray, B.K. and Ray, K.S., 1995. Corner detection using iterative Gaussian smoothing with constant window size. Pattern Recognition. 28 (11), 1765-1781.

[Ray, B.K. and Ray, K., 1992] Ray, B.K. and Ray, K., 1992. An algorithm for detection of dominant points and polygonal approximation of digitized curves. Pattern Recognition Letters. 13(12), 849-856.

[Rohr, 1997] Rohr, K., 1997. On 3D differential operators for detecting point landmarks. Image and Vision Computing, Elsevier. 15(3), 219-233.

[Rohr, 1999] Rohr, K., 1999. Extraction of 3d anatomical point landmarks based on invariance principles. Pattern Recognition, Elsevier Science, Oxford, U.K. 32(1), 3-15.

[Rosenfeld and Johnston, 1973] Rosenfeld, A. and Johnston, E., 1973. Angle Detection on Digital Curves. IEEE Transactions on Computers. 22, 875-878.

[Rosenfeld and Weszka, 1975] Rosenfeld, A. and Weszka, J.S., 1975. An improved method of angle detection on digital curves. IEEE Transactions on Computers. 24(9), 940-941.

[Rueckert et al., 2001] Rueckert, D., Frangi, A., Schnabel, J., 2001. Automatic construction of 3D statistical deformation models using non-rigid registration. Proceedings of MICCAI01. 77-84.

[Rueda et al., 2008a] Rueda, S, Udupa, J.K, and Bai, L, 2008. Local curvature scale: a new concept of shape description. Medical Imaging 2008: Image Processing. Edited by Reinhardt, J.M, Pluim, J.P.W. Proceedings of SPIE, SPIE, Bellingham, WA, USA. 6914, 69144Q.

[Rueda et al., 2008b] Rueda, S., Udupa, J.K., and Bai, L., 2008. Landmark Selection for Shape Model Construction Via Equalization of Variance. IEEE International Symposium on Biomedical Imaging (ISBI) 2008: from Nano to Macro. 1271-1274.

[Saha et al., 2000] Saha, P., Udupa, J.K., and Odhner, D., 2000. Scale-based fuzzy connected image segmentation: theory, algorithms, and validation. Computer Vision and Image Understanding, Academic Press. 77(2), 145-174.

[Serra, 1982] Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press, London, U.K.

[Sonka et al., 1999] Sonka, M., Hlavac, V., and Boyle, R., 1999. Image Processing, Analysis, and Machine Vision. PWS Publishing, New York, USA.

[Souza and Udupa, 2005] Souza, A. and Udupa, J.K., 2005. Automatic landmark selection for Active Shape Models. Medical Imaging 2005: Image Processing. Edited by Fitzpatrick, J.M. and Reinhardt, J.M. Proceedings of SPIE, SPIE, Bellingham, WA, USA. 5747, 1377-1383.

[Squire and Caelli, 2000] Squire, D. M. and Caelli, T. M., 2000. Invariance signatures: characterizing contours by their departures from invariance. Computer Vision and Image Understanding, Academic Press. 77(3), 284-316.

[Styner et al., 2003] Styner, M.A., Rajamani, K.T., Nolte, L., Zsemlye, G., Székely, G., Taylor, C.J., and Davies, R.H., 2003. Evaluation of 3d correspondence methods for model building. Proceedings in Information Processing in Medical Imaging. LNCS 2732, 63-75.

[Taza and Suen, 1989] Taza, A. and Suen, C., 1989. Discrimination of planar shapes using shape matrices. IEEE Transactions on systems, man, and cybernetics. 19, 1281-1289.

[Thodberg, 2003a] Thodberg, H.H., 2003. Minimum description length shape and appearance models. Proceedings in Information Processing in Medical Imaging. LNCS 2732, 51-62.

[Thodberg, 2003b] Thodberg, H.H. and Olafsdottir, H., 2003. Adding curvature to minimum description length shape models. 14th British Machine Vision Conference. 2, 251-260.

[Thompson, 1997] Thompson, P. and Toga, A., 1997. Detection, visualization and animation of abnormal anatomic structure with a probabilistic brain atlas based on random vector field transformations. Medical Image Analysis. 1, 271-294.

[Tieng and Boles, 1997] Tieng, Q.M. and Boles, W.W., 1997. Recognition of 2D object contours using the wavelet transform zero-crossing representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 19(8), 910-916.

[Torres et al., 2004] Torres, R. da S., Falcão, A.X., Costa, L. da C., 2004. A graph-based approach for multiscale shape analysis. Pattern Recognition, Elsevier Science, Oxford, U.K. 37(6), 1163-1174.

[Twining et al., 2002] Twining, C., Marsland, S., Taylor, C., 2002. Measuring geodesic distances on the shape of bounded diffeomorphisms. 13^{th} British Machine Vision Conference, BMVA. 2, 847-856.

[Udupa, 1994] Udupa, J.K., 1994. Multidimensional digital boundaries. CVGIP: Graphical Models and Image Processing. 56(4), 311-323.

[Vaillant et al., 2004] Vaillant, M., Miller, M., Younes, L., and Trouve, A., 2004. Statistics on diffeomorphisms via tangent space representations. Neuroimage. 23, S161-S169.

[Walker et al., 2002] Walker, K.N., Cootes, T.F., and Taylor, C.J., 2002. Automatically Building Appearance Models from Image Sequences using Salient Features. Image and Vision Computing, Elsevier. 20(5-6), 435-440.

[Witkin, 1983] Witkin, A.P., 1983. Scale-Space Filtering. International Joint Conference on Artificial Intelligence. 1019-1022.

[Worring and Smeulders, 1992] Worring, M. and Smeulders, A.W.M., 1992. The accuracy and precision of curvature estimation methods. International Conference on Pattern Recognition. III, 139-142.

[Yang et al., 1998] Yang, H.S., Lee, S.U., and Lee, K.M., 1998. Recognition of 2D object contours using starting-point-independent wavelet coefficient matching. Journal of Visual Communication and Image Representation. 9(2), 171-181.

[Zahn and Roskies, 1972] Zahn, C.T. and Roskies, R.Z., 1972. Fourier descriptors for plane closed curves. IEEE Transactions on Computers. C-21 (3), 269-281.

[Zhang and Lu, 2004] Zhang, D. and Lu, G., 2004. Review of shape representation and description techniques. Pattern recognition. 37(1), 1-19.

[Zhang and Lu, 2002a] Zhang, D.S. and Lu, G., 2002. Generic Fourier descriptor for shape-based image retrieval. Proceedings of IEEE International Conference on Multimedia and Expo (ICME2002), Lausanne, Switzerland. 1, 425-428.

[Zhang and Lu, 2002b] Zhang, D.S. and Lu, G., 2002. Enhanced Generic Fourier descriptor for object-based image retrieval. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002). 4, 3668-3671.